REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-98-

$\mathcal{D}$778

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the t ring and maintaining the data needed, and completing and reviewing the collection of information. Send comments i n of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for ir uite 1204, Arlington, VA 22202-4302, and to the Office of management and Budget, Paperwork Reduction Project (0704-(

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE December, 1993	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE
USAF Summer Research Program - 1993 Graduate Student Research Program Final Reports, Volume 11, AEDC, FJSRL, and WHMC

5. FUNDING NUMBERS

6. AUTHORS
Gary Moore

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Research and Development Labs, Culver City, CA

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NI
4040 Fairfax Dr, Suite 500
Arlington, VA 22203-1613

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Contract Number: F4962-90-C-0076

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The United States Air Force Graduate Student Research Program (USAF- GSRP) is designed to introduce university, college, and technical institute graduate students to Air Force research. This is accomplished by the graduate students being selected on a nationally advertised competitive basis during the summer intersession period to perform research at Air Force Research Laboratory Technical Directorates and Air Force Air Logistics Centers. Each participant provided a report of their research, and these reports are consolidated into this annual report.

Reproduced From
Best Available Copy

14. SUBJECT TERMS
AIR FORCE RESEARCH, AIR FORCE, ENGINEERING, LABORATORIES, REPORTS, SUMMER, UNIVERSITIES

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239.18
Designed using WordPerfect 6.1, AFOSR/XPP, Oct 96

UNITED STATE AIR FORCE

SUMMER RESEARCH PROGRAM -- 1993

SUMMER RESEARCH PROGRAM FINAL REPORTS

VOLUME 11

ARNOLD ENGINEERING DEVELOPMENT CENTER
FRANK J. SEILER RESEARCH LABORATORY
WILFORD HALL MEDICAL CENTER

RESEARCH & DEVELOPMENT LABORATORIES

5800 Uplander Way

Culver City, CA 90230-6608

Program Director, RDL
Gary Moore

Program Manager, AFOSR
Col. Hal Rhoades

Program Manager, RDL
Scott Licoscos

Program Administrator, RDL
Gwendolyn Smith

Program Administrator, RDL
Johnetta Thompson

Submitted to:

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Bolling Air Force Base

Washington, D.C.

December 1993

19981211 019

Master Index For Graduate Students

Ahmad, Imad
BS
Electrical Engineering
Portland State University
Portland, OR 97207-0000

Field: Electrical Engineering
Laboratory: PL/WS

Vol-Page No: 8-19

Anselmo, Andrew
MS
Mechanical Engineering
Columbia University
New York, NY 10027-0000

Field: Mechanical Engineering
Laboratory: RL/ER

Vol-Page No: 9- 3

Bapty, Theodore
MS
Box 1649 Station B
Vanderbilt University
Nashville, TN 37235-0000

Field: Electrical Engineering
Laboratory: AEDC/

Vol-Page No: 11- 1

Barber, Brian
BS
Electrical Engineering
Washington University
St. Louis, MO 63130-0000

Field: Electrical Engineering
Laboratory: RL/OC

Vol-Page No: 9-10

Bard, David
MS
Mechanical Engineering
Columbia University
New York, NY 10027-0000

Field: Applied Mechanics
Laboratory: PL/SX

Vol-Page No: 8-14

Barnett, William
BS
CLOUD & Aerosol Science L
University of Missouri
Rolla, MO 65401-0000

Field: Chemistry
Laboratory: PL/LI

Vol-Page No: 8- 6

Bartell, Craig
BS
Chemistry
Wright State University
Fairborn, OH 45435-0000

Field: Chemistry
Laboratory: WL/PO

Vol-Page No: 10-26

Bartelt, Timothy
MS
Physics
Clarkson University
Potsdam, NY 13676-0000

Field: Physics
Laboratory: RL/ER

Vol-Page No: 9- 4

Brocklehurst, William
BS
Aerospace Engineering
University of Cincinnati
Cincinnati, OH 45221-0000

Field: Aerospace Engineering
Laboratory: PL/RK

Vol-Page No: 8-10

Burns, Paul
BS
Electrical Engineering
Auburn University
Auburn, AL 36849-0000

Field: Electrical Engineering
Laboratory: WL/MN

Vol-Page No: 10-21

Carlen, Edwin
BS
Electrical Engineering
Oakland University
Rochester, MI 48309-4401

Field: Electrical Engineering
Laboratory: WL/AA

Vol-Page No: 10- 1

Carter, Charity
MS
Electrical Engineering
Stevens Institute of Tech.
Hoboken, NJ   7030-0000

Field: Interdisciplinary Engr.
Laboratory: RL/IR

Vol-Page No: 9- 8

Caslin, Barry
BS
Computer Science/Engineer
Wright State University
Dayton, OH 45435-0000

Field: Computer Science
Laboratory: WL/ML

Vol-Page No: 10-14

Charley, David
BS
Electrical Engineering
University of Cincinnati
Cincinnati, OH 45221-0030

Field: Computer Science
Laboratory: WL/AA

Vol-Page No: 10- 2

Cisneros, John
BS
Psychology
California State University
Los Angeles, CA 90032-0000

Field: Psychology
Laboratory: RL/XP

Vol-Page No: 7- 8

Craig, Ralph
BS
Mathematics
North Carolina State Univ.
Raleigh, NC 27695-8205

Field: Mathematics
Laboratory: PL/VT

Vol-Page No: 8-15

GSRP Participant Data

Curtis, Joseph
MS
Zoology
University of California
Davis, CA 95616-0000

Field: Biomedicine
Laboratory: AL/OE

Vol-Page No: 7-25

Darnell, Julie
BS
Mechanical Engineering
University of Tennessee
Tullahoma, TN 37388-0000

Field: Mechanical Engineering
Laboratory: AEDC/

Vol-Page No: 11- 2

Davis, Wyatt
BS
Mechanical Engineering
Washington State University
Pullman, WA 99164-2920

Field: Mechanical Engineering
Laboratory: FJSRL/

Vol-Page No: 11-10

DeVilbiss, Alan
BS
Physics
University of Colorado
Colorado Springs, CO 80901-7150

Field: Physics
Laboratory: FJSRL/

Vol-Page No: 11-11

Detwiler, Duane
BS
Aeronautical Engineering
Ohio State University
Columbus, OH 43235-0000

Field: Aerospace Engineering
Laboratory: WL/FI

Vol-Page No: 10- 7

DiPietro, Jr., Anthony
BS
Aerospace Engineering
Virginia Polytechnic Institute
Blacksburg, VA 24060-0000

Field: Aerospace Engineering
Laboratory: AEDC/

Vol-Page No: 11- 3

Dobransky, Mary
MS
Systems Science
Binghamton University
Binghamton, NY 13902-6000

Field: Advanced Technology
Laboratory: RL/XP

Vol-Page No: 9-15

Doss, Ellen
BS
Biophysics
University of Scranton
Scranton, PA 18505-0898

Field: Biophysics
Laboratory: AL/AO

Vol-Page No: 7- 1

GSRP Participant Data

DuBois, Travis
BS
Mechanical Engineering
Old Dominion University
Norfolk, VA 23529-0247

Field: Mechanical Engineering
Laboratory: FJSRL/

Vol-Page No: 11-12

Farquhar, John
MS
Instructional Technology
University of Georgia
Athens, GA 30602-0000

Field: Instructional Technology
Laboratory: AL/HR

Vol-Page No: 7-17

Fischer, Verlyn
BS
Materials Science & Engr.
University of Florida
Gainesville, FL 32611-0000

Field: Physics
Laboratory: WL/ML

Vol-Page No: 10-15

Foster, Robert
BS
Mechanical Engineering
University of Iowa
Iowa City, IA 52240-0000

Field: Mechanical Engineering
Laboratory: WL/PO

Vol-Page No: 10-27

Frank, Jonathan
MS
Mechanical Engineering
Yale University
New Haven, CT   6520-0000

Field: Mechanical Engineering
Laboratory: WL/PO

Vol-Page No: 10-28

Fuller, Joan
BS
Chemistry
University of Alabama
Tuscaloosa, AL   -   0

Field: Chemistry
Laboratory: FJSRL/

Vol-Page No: 11-13

Gavora, Mark
MS
Instructional Systems
Florida State University
Tallahassee, FL 32304-3551

Field: Instructional Systems
Laboratory: AL/HR

Vol-Page No: 7-18

Geierman, Robert
BS
Mechanical Engineering
University of Tennessee
Tullahoma, TN 37388-0000

Field: Mechanical Engineering
Laboratory: AEDC/

Vol-Page No: 11- 4

GSRP Participant Data

Gluck, Kevin
BS
Psychology
Trinity University
San Antonio, TX 78212-0000

Field: Psychology
Laboratory: AL/HR

Vol-Page No: 7-19

Gottlob, Lawrence
MS
Psychology
Arizona State University
Tempe, AZ 85287-0000

Field: Psychology
Laboratory: AL/HR

Vol-Page No: 7-20

Griffin, Steven
BS
Engineering
University of Texas
San Antonio, TX 78249-0665

Field: Mechanical Engineering
Laboratory: PL/VT

Vol-Page No: 8-16

Grosskopf, Kevin
BS
FAC 101
University of Florida
Gainesville, FL 32611-0000

Field: School of Bldg. Construct
Laboratory: WL/FI

Vol-Page No: 10- 8

Harn, Patricia
BS
Technical Communication
University of Washington
Seattle, WA 98195-0000

Field: Communications
Laboratory: AL/HR

Vol-Page No: 7-21

Hickman, Mary
BS
Physics and Astronomy
University of Wyoming
Laramie, WY 82071-0000

Field: Astrophysics
Laboratory: PL/LI

Vol-Page No: 8- 7

Hovey III, Leland
BS
Computer Science
SUNY Institute of Technology
Utica, NY 13504-3050

Field: Mathematics
Laboratory: RL/XP

Vol-Page No: 9-16

Jain, Sanjay
BS

Northeastern Ohio University
Rootstown, OH 44272-0000

Field: Natural Sciences
Laboratory: WHMC/

Vol-Page No: 11-16

Jefferson, Ellen
BS
Biology
Trinity University
San Antonio, TX 78212-0000

Field: Biology
Laboratory: AL/OE

Vol-Page No: 7-26

Jenny, Jason
BS
Materials Science
Carnegie-Mellon University
Pittsburgh, PA 15213-0000

Field: Materials Science
Laboratory: WL/ML

Vol-Page No: 10-16

Jolly, Mohanjit
BS
Aeronautics/Astronautics
MIT
Cambridge, MA   2139-0000

Field: Aeronautics/Astronautics
Laboratory: PL/RK

Vol-Page No: 8-11

Jones, Claud
BS
Electrical & Computer Eng
University of SW Louisiana
Lafayette, LA 70504-3890

Field: Petroleum Engineering
Laboratory: RL/C3

Vol-Page No: 9- 1

Kelly, John
BS
Electrical Engineering
University of Florida
Gainesville, FL 32601-0000

Field: Electrical Engineering
Laboratory: PL/LI

Vol-Page No: 8- 8

Kimball, Lucia
MS
Mathematics
Worcester Polytechnic Inst.
Worcester, MA   1609-0000

Field: Applied Mathematics
Laboratory: PL/GP

Vol-Page No: 8- 1

Kundich, Robert
MS
899 Madison Ave., Ste 801
Univ. Tennessee-Memphis
Memphis, TN 38163-0000

Field: Biomedical Engineering
Laboratory: AL/CF

Vol-Page No: 7- 9

Kvasnak, William
MS
Mechanical Engineering
Clarkson University
Potsdam, NY 13699-5725

Field: Mechanical Engineering
Laboratory: AEDC/

Vol-Page No: 11- 5

Lair, John
BS
Civil Engineering
University of New Orleans
New Orleans, LA 70148-0000

Field: Industrial Design
Laboratory: WL/FI

Vol-Page No: 10- 9

Lee, Daniel
BS
Electrical Engineering
Southern Illinois University
Carbondale, IL 62901-0000

Field: Electrical Engineering
Laboratory: RL/ER

Vol-Page No: 9- 5

Leiweke, Robert
BS
Aeronautical Engineering
Ohio State University
Columbus, OH 43210-0000

Field: Aerospace Engineering
Laboratory: PL/WS

Vol-Page No: 8-20

Litvin, Kerry
MS
Electrical Engineering
Cornell University
Ithaca, NY 14853-0000

Field: Electrical Engineering
Laboratory: RL/OC

Vol-Page No: 9-11

Luker, Stephen
MS
Biological Science
University of Alabama
Tuscaloosa, AL 35487-0344

Field: Marine Science
Laboratory: PL/GP

Vol-Page No: 8- 2

Masterson, Gina
BS
Industrial Engineering
Auburn University
Auburn, AL   -   0

Field: Industrial Engineering
Laboratory: AL/OE

Vol-Page No: 7-27

McMurtry, J.
MS
Mechanical Engineering
Louisiana Tech University
Ruston, LA 71272-0046

Field: Mechanical Engineering
Laboratory: WL/MN

Vol-Page No: 10-22

Menendez-Barreto, Melani
MS
Physics
University of Puerto Rico
Mayaguez, PR   681-0000

Field: Physics
Laboratory: PL/GP

Vol-Page No: 8- 3

GSRP Participant Data

Messerschmitt, Jane
MS
Electrical Engineering
Polytechnic University
Farmingdale, NY 11735-0000

Field:
Laboratory: PL/WS

Vol-Page No: 8-21

Mills, Jeffrey
MS
Cheistry
Indiana University
Bloomington, IN 47405-0000

Field: Chemistry
Laboratory: PL/RK

Vol-Page No: 8-12

Monka, Gary
BS
Chemistry
University of Scranton
Scranton, PA 18510-0000

Field: Chemistry
Laboratory: AL/OE

Vol-Page No: 7-28

Moore, Michael
MS
Electrical Engineering
Vanderbilt University
Nashville, TN 37235-0000

Field: Electrical Engineering
Laboratory: AEDC/

Vol-Page No: 11- 6

Moriarty, Daniel
MS
Nuclear Engineering
MIT
Cambridge, MA   2139-0000

Field: Nuclear Engineering
Laboratory: PL/GP

Vol-Page No: 8- 4

Nagaraja, Chandra
BS
Mechanical Engineering
University of Dayton
Dayton, OH 45469-0000

Field: Mechanical Engineering
Laboratory: WL/PO

Vol-Page No: 10-29

Naghski, David
MS
Electrical Engineering
University of Cincinnati
Cincinnati, OH 45221-0000

Field: Electrical Engineering
Laboratory: WL/ML

Vol-Page No: 10-17

Neaffer, Ronald
BS
Polymer Science
University of Akron
Akron, OH 44325-3909

Field: Physics
Laboratory: WL/ML

Vol-Page No: 10-18

GSRP Participant Data

Newell, Tim
MS
Physics
University of North Texas
Denton, TX 76203-0000

Field: Physics
Laboratory: PL/LI

Vol-Page No: 8- 9

Nguyen, Sonny
BS
Electrical Engineering
Wright University
Dayton, OH 45401-0000

Field: Electrical Engineering
Laboratory: WL/PO

Vol-Page No: 10-30

Nuteson, Todd
BS
Electrical Engineering
Wright State University
Dayton, OH 45435-0000

Field: Electrical Engineering
Laboratory: WL/EL

Vol-Page No: 10- 4

Olson, Michael
BS
Electrical Engineering
Florida State University
Tallahassee, FL 32316-0000

Field: Electrical Engineering
Laboratory: WL/MN

Vol-Page No: 10-23

Panek, Heather
MS
Biology
University of Scranton
Scranton, PA 18510-0000

Field: Biochemistry
Laboratory: AL/AO

Vol-Page No: 7- 3

Petroziello, Joseph
BS
Biology
University of Scranton
Scranton, PA 18510-0000

Field: Biology
Laboratory: AL/AO

Vol-Page No: 7- 4

Pobst, Jeffrey
BS
Aerospace Engineering
University of Southern Calif.
Los Angeles, CA 90089-1191

Field: Aerospace Engineering
Laboratory: PL/RK

Vol-Page No: 8-13

Povich, Clinton
BS
Aerospace Engineering
University of Kansas
Lawrence, KS 66045-0000

Field: Aerospace Engineering
Laboratory: FJSRL/

Vol-Page No: 11-14

GSRP Participant Data

Radomsky, Leon
MS
Chemical Engineering
Columbia Univeristy
New York, NY 10027-0000

Field: Materials Science
Laboratory: WL/EL

Vol-Page No: 10- 5

Reed, Jason
BS
Electrical Engineering
Cornell University
Ithaca, NY 14853-0000

Field: Physics
Laboratory: RL/OC

Vol-Page No: 9-12

Reichmeyer, Francis
MS
Electlrical Engineering
Syracuse University
Syracuse, NY 13244-1240

Field: Electrical Engineering
Laboratory: RL/C3

Vol-Page No: 9- 2

Reiley, Daniel
MS
Physics
University of Alabama
Huntsville, AL 35899-0000

Field: Physics
Laboratory: WL/MN

Vol-Page No: 10-24

Reister, Craig
BS
Electrical Engineering
North Dakota State University
Fargo, ND 58105-0000

Field: .Electrical Engineering
Laboratory: AL/AO

Vol-Page No: 7- 5

Reuster, Daniel
MS
Electrical Engineering
University of Dayton
Dayton, OH 45469-0000

Field: Electrical Engineering
Laboratory: WL/AA

Vol-Page No: 10- 3

Riesinger, Joyce
BS
Ecology/Evolutionary Biol
University of West Florida
Pensacola, FL 32514-0000

Field: Marine Biology
Laboratory: AL/EQ

Vol-Page No: 7-14

Ryan, Arthur
BS
Psychology
Wright State University
Dayton, OH 45435-0001

Field: Psychology
Laboratory: AL/CF

Vol-Page No: 7-10

Samad, Mohammed
BS
Mechanical Engineering
University of New Orleans
New Orleans, LA 70148-0000

Field: Mechanical Engineering
Laboratory: WL/FI

Vol-Page No: 10-10

Schroeder, Mark
BS
Electrical Engineering
North Dakota State University
Fargo, ND 58105-0000

Field: Electrical Engineering
Laboratory: AL/AO

Vol-Page No: 7- 6

Schwing, Ronald
BS
Mechanical Engineering
University of Iowa
Iowa City, IA 52242-0000

Field: Mechanical Engineering
Laboratory: WL/PO

Vol-Page No: 10-31

Seifert, Jason
BS
Psychology
University of Dayton
Dayton, OH 45410-0000

Field: Psychology
Laboratory: AL/HR

Vol-Page No: 7-22

Seydel III, Robert
BS
Aerospace Engineering
University of Missouri
Rolla, MO 65401-0000

Field: Aerospace Engineering
Laboratory: WL/FI

Vol-Page No: 10-11

Shahidi, Anoosh
MS
Intelligent Systems Studi
University of Pittsburgh
Pittsburgh, PA 15260-0000

Field: Intelligent Systems
Laboratory: AL/HR

Vol-Page No: 7-23

Shahrokhi, Kimball
BS
Mechanical Engineering
Vanderbilt University
Nashville, TN 37232-0000

Field: Mechanical Engineering
Laboratory: AEDC/

Vol-Page No: 11- 7

Slater III, Robert
MS
Mechanical Engineering
University of Cincinnati
Cincinnati, OH 45221-0000

Field: Engineering
Laboratory: WL/FI

Vol-Page No: 10-12

GSRP Participant Data

Socci, Ed
MS
Materials Science & Engr
University of Virginia
Charlottesville, VA 22903-2442

Field:
Laboratory: WL/ML

Vol-Page No: 10-19

Sodoy, Maureen
BS
Biology
Trinity University
San Antonio, TX 78212-0000

Field: Biology
Laboratory: AL/OE

Vol-Page No: 7-29

Sorensen, Bryant
MS
Electrical Engineering
University of Tennessee
Tullahoma, TN 37388-0000

Field: Mathematics
Laboratory: AEDC/

Vol-Page No: 11- 8

Spencer, Robert
BS
Electrical Engineering
Cornell University
Ithaca, NY 14853-0000

Field: Electrical Engineering
Laboratory: RL/OC

Vol-Page No: 9-13

Stadler, Bethanie
BS
Materials Science
Massachusetts Inst. of Tech.
Cambridge, MA   2139-0000

Field: Materials Science
Laboratory: RL/ER

Vol-Page No: 9- 6

Stansbery, Donald
BS
Aerospace Engineering
University of Missouri
Rolla, MO 65401-0000

Field: Aerospace
Laboratory: WL/MN

Vol-Page No: 10-25

Starchville, Jr., Thomas
MS
Aerospace Engineering
Pennsylvania State University
University Park, PA 16802-0000

Field: Aerospace Engineering
Laboratory: WL/FI

Vol-Page No: 10-13

Starks, Michael
BS
Electrical Engineering
Boston University
Boston, MA   2215-0000

Field: Electrical Engineering
Laboratory: PL/GP

Vol-Page No: 8- 5

GSRP Participant Data

Stauffer, Joseph
MS
Management
University of Iowa
Iowa City, IA 52242-1323

Field: Management
Laboratory: AL/HR

Vol-Page No: 7-24

Stenger, Vincent
BS
Electrical Engineering
University of Cincinnati
Cincinnati, OH 45221-0030

Field: Electrical Engineering
Laboratory: WL/EL

Vol-Page No: 10- 6

Stohs, Jonathan
BS
Physics
University of New Mexico
Albuquerque, NM 87131-0000

Field: Math
Laboratory: PL/VT

Vol-Page No: 8-17

Stroman, Ronald
BS
Pharmacy
University of South Carolina
Columbia, SC 29208-0000

Field: Pharmacy
Laboratory: AL/AO

Vol-Page No: 7- 7

Sullins, Tashia
BS
Chemistry
University of Georgia
Athens, GA 30605-0000

Field: Chemistry
Laboratory: AL/EQ

Vol-Page No: 7-15

Sutcliffe, Ronald
BS
Chemistry
Southwest Texas State
San Marcos, TX 78666-0000

Field: Chemistry
Laboratory: AL/OE

Vol-Page No: 7-30

Sweeney, Walter
MS
Electrical Engineering
University of Maine
Orono, ME   4469-0000

Field: Electrical Engineering
Laboratory: RL/IR

Vol-Page No: 9- 9

Swindal, J.
MS
Applied Physice
Yale University
New Haven, CT   6520-0000

Field: Applied Physics
Laboratory: WL/PO

Vol-Page No: 10-32

GSRP Participant Data

Tascillo, Mark
MS
Electrical Engineering
Binghamton University
Binghamton, NY 13902-0000

Field: Electrical Engineering
Laboratory: RL/OC

Vol-Page No: 9-14

Thomas, Stuart
BS
Chemical Engineering
North Carolina State Univ.
Raleigh, NC 27695-0000

Field: Chemical Engineering
Laboratory: AL/EQ

Vol-Page No: 7-16

Tipton, Kevin
MS
Nutrition and Food Scienc
Auburn University
Auburn, AL 36849-5605

Field: Zoology
Laboratory: AL/CF

Vol-Page No: 7-11

Tornow, Carina
BS
Quantitive Psychology
Ohio State University
Columbus, OH 43210-0000

Field: Psychology
Laboratory: AL/CF

Vol-Page No: 7-12

Trick, Kimberly
MS
Chemical Engineering
University of Dayton
Dayton, OH 45401-0240

Field: Chemical Engineering
Laboratory: WL/ML

Vol-Page No: 10-20

Weaver, Michael
MS
Aerospace Engineering
Georgia Institute of Technolog
Atlanta, GA 30332-0150

Field: Aerospace Engineering
Laboratory: AEDC/

Vol-Page No: 11- 9

Weitzman, Peter
MS
Thayer School of Engineer
Dartmouth College
Hanover, NH 3755-0000

Field: Electrical Engineering
Laboratory: FJSRL/

Vol-Page No: 11-15

Whitmore, Christine
BS
Mechanical Engineering
Washington University
St. Louis, MO   -   0

Field: Mechanical Engineering
Laboratory: AL/CF

Vol-Page No: 7-13

Widener, Charles
MS
Electrical Engineering
Syracuse University
Syracuse, NY 13244-0000

Field: Electrical Engineering
Laboratory: PL/VT

Vol-Page No: 8-18

Wierman, Mark
MA
Systems Science
Binghamton University
Binghamton, NY 13902-6000

Field: Mathematics
Laboratory: RL/XP

Vol-Page No: 9-17

Willemsen, Balam
MS
Physics
Northeastern University
Boston, MA   2115-0000

Field: Physics
Laboratory: RL/ER

Vol-Page No: 9- 7

Wolfe, Randall
MS
Biopsychology
University of Georgia
Athens, GA 30612-0000

Field: General Psychology
Laboratory: AL/OE

Vol-Page No: 7-31

Worthy, Mark
BS
Environmental Engineering
University of Alabama
Huntsville, AL 35899-1750

Field: Electrical Engineering
Laboratory: PL/WS

Vol-Page No: 8-22

PARALLEL TURBINE ENGINE INSTRUMENTATION SYSTEM

Ted A. Bapty

Graduate Student

Department of Electrical Engineering

Vanderbilt University

400 24th Avenue South

Nashville, TN 37235

PARALLEL TURBINE ENGINE INSTRUMENTATION SYSTEM

Ted A. Bapty

Research Faculty

Department of Electrical Engineering

Abstract

The complexity of modern turbine engines requires significant amounts of testing to ensure correct, cost effective development. Ground testing enables highly accurate control over the external conditions along with the ability to acquire a large number of measurements of internal structures.

Analysis of turbine engine testing data is a computationally intensive, dynamic process. The combination of high bandwidths, large numbers of channels, and constantly changing processing requirements place many demands on an instrumentation and data analysis system.

The Computer Assisted Dynamic Data Acquisition and Monitoring System(CADDMAS), developed in conjunction with Arnold Engineering Development Center, has been in operation for several months in support of simulated altitude turbine engine stress testing. The system uses parallel processing to sustain over 400 million floating point operations per second(MFLOPS) producing on-line data analysis plots, such as Campbell Diagrams, Phase Campbell Diagrams, and spectral energy plots. Until recently, these plots took several days to weeks to obtain. The plots are displayed on multiple high speed color displays using standard engineering graphics. Hardcopy plots are available on demand.

I describe hardware and software architecture of a greater than 100 processor parallel instrumentation system for turbine engine aeromechanical stress analysis. The hardware consists of a mixture of hetereogeneous processing nodes arranged in a configuration closely matching the analysis algorithms. Specialized processors are used for numerical, input/output, graphics, user interface, and storage.

The software uses model-based program synthesis techniques to dynammically reconfigure signal analysis algorithms to meet changing requirements.

PARALLEL TURBINE ENGINE INSTRUMENTATION SYSTEM

Ted A. Bapty

Introduction

During the development and operational lifespan of a turbine engine design numerous tests are performed to ensure design integrity and performance over time. Ground testing forms a basis for this testing. The engine is installed in a controlled environment test cell to simulate a wide variety of conditions. The conditions are adjusted to simulate the entire operating envelope of the engine, to predict the operation of the engine in the field.

Turbine engine stress testing is a critical phase in the testing process. In this test mode, the engine is instrumented with hundreds of sensors measuring the stress on various internal components. The stress sensors are mounted on blades and stators, directly measuring dynamic vibrations. The signals are telemetered out of the engine into the data acquisistion system. In addition to these signals, a wide variety of external parameters such as airflow, temperature, pressure, etc, are measured.

Turbine Engine testing generates massive volumes of data at very high rates. A single engine may have several hundred sensors, with bandwidths up to tens of kilohertz. Test periods last for several hours. The entire bulk of data generated is in the terabytes for a single test.

Complete analysis of all acquired data has been technically impossible in the past, due to the speed and capacity of standard computers. Historically, only a small fraction of the data has been fully processed by periodic sampling of the data stream. This analysis was also performed off-line, with delays of days to weeks before results were available. This latency prohibits the use of results to guide the testing process.

On-line processing of these signals is critical to improving the testing process. The results from the on-line computations should be available to enable mid-test decisions, interactive test planning. Furthermore, on-line examination of information can assist in catastrophe avoidance, by enabling the detection of pathological engine conditions.

Performing these computations and displaying the results in an interactive format requires an extremely high processing rate. Typical algorithms include frequency transformation, peak detection and validation and correlation. For the analysis of 24 channels of stress data, approximately 400 MFLOPS of sustained computation is required. To process the hundreds of channels available in a typical test, an aggregate sustained computation rate in the GFLOPS range is necessary to meet requirements. Note that these are sustained computational rates, not to be confused with Peak MFLOPS advertised by computer manufacturers. The situation is further exacerbated by the requirement to acquire the volumes of data into the computer and to plot the results quickly. The aggregate data input is in the tens of MWords per second. Compounding these requirements, is the real-time requirement that the system must keep up with the data stream and display results interactively.

Typical supercomputers advertise processing speeds in these ranges. Some support data acquisition systems capable of meeting the input bandwidth requirements. The problem with applying standard supercomputers to this type of task is that both of these facilities are typically not available concurrently. The computer must simultaineously accept and process the data. Supercomputers also tend to be expensive to acquire and maintain, especially with high-bandwidth acquisition systems. The real-time nature is complicated by the batch-oriented operating systems typically found on supercomputers.

The majority of algorithms required for turbine engine testing are applied independently on each of the channels. The high computational requirements and natural parallelism make parallel processing a cost-effective hardware approach. The data streams are physically separate channels, so a parallel I/O system is also applicable.

To achieve the multi-GFLOPS goal, a large number of processors must be applied. I designed and built a parallel processing system based on the Inmos Transputer to process turbine engine stress data on-line. The bare Transputer however, lacks the computational capabilities required for this task. At 2 MFLOPS (Peak) execution speed, over 1000 processors would be required to achieve the 2 GFLOPS needed. The I/O capabilities for analog input and graphics output are not inherent in the chip, so these features are added, producing a heterogeneous architecture. In the section 2, I will discuss the hardware configuration.

While the hardware is available (most of the components are available commercially) and the technology allows easy configurations of arbitrary topologies, the software technology is not as simple. The software is complicated by several factors:

1. **Distributed Control:** Instead of having one thread of control executing in the system, I have as many threads as there are processors. The problem lies in getting all of these threads to cooperate and accomplish a single task.

2. **Communication:** As the computational data dependencies are divided across processors, there must be a method for transmitting intermediate results from one place to another.

3. **Real-Time:** The data is coming into the system at a constant rate. The system must respond to all of the data at this rate, or data will be lost.

4. **Dynamically Changing Requirements:** The processing requested from the system is constantly changing. Each test point has a different mix of required output processing.

5. **Modularity:** Systems of various sizes and processing capacity are required. The components (hardware and software) must be designed to allow modules to be plugged together to quickly and easily form arbitrary systems.

These problems must be solved in a way that will not make the system implementation cost-prohibitive to implement. In the Software Environment section I discuss the software methodology used to implement the parallel instrumentation systems such as the CADDMAS 24 channel prototype system.

CADDMAS Hardware Architecture

The CADDMAS system is a heterogeneous, massively parallel system designed for parallel instrumentation applications. The current prototype relies on the Inmos Transputer for the basic processing nodes of the architecture.

A Transputer is used in conjunction with each of the processing nodes. It provides the general message passing and control fabric of the system. The basic nodes are augmented as necessary to get the necessary computational characteristics. The basic nodes are as follows:

1. *The Transputer Processing Element* uses a *T800* and dynamic RAM (1 or 4 MBytes) for general purpose data manipulation tasks, such as data accumulation and plotter display list generation.
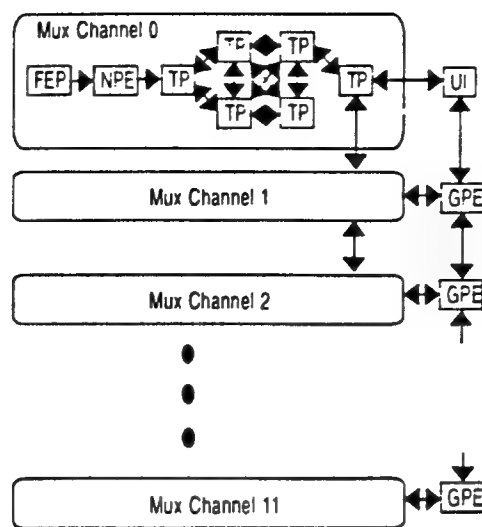
Figure 1: Real-Time Data Visualization System

2. *The Numeric Processing Element* uses a *T800* and a *Zoran floating point Vector Signal Processor* to supply the bulk of the floating point computational horsepower.

3. *The Graphics Processing Element* consists of a IBM/386 compatible personal computer with an accelerated graphics card. The PC also provides interfaces for mouse and keyboard input and limited digital storage.

4. *Front End Processor* samples the analog data, and detects peaks, filters and downsamples the digital data.

The CADDMAS 24 channel prototype application of these processors to on-line test data processing requires a mix of approximately 100 processors. Figure 1 shows a typical full-scale system.

The next two subsections describe in further detail the custom-designed components. For detailed descriptions of the commercially-available components, please refer to the product literature.

Numeric Processing Element

The Numeric Processing Element uses a 20 MHz T800 for communications and control. The Transputer has access to 1024K Bytes of static memory. The Zoran Vector Signal (VSP) Processor can also be a bus master to access this memory.

The VSP is modeled after an array processor architecture. The VSP has high speed internal memory to allow concurrent processing with the host T800. Internal organization of the VSP allows concurrent operation within the subsystem, to further increase processing speed.

The VSP implements IEEE standard 32 bit floating point. Its instruction set includes high-level routines that support common signal processing tasks. Vector addition, multiplication, polynomial expansion, min/max, accumulate, int to FP, and FP to int are single instructions. There are also instructions that implement FIR and IIR filtering. Frequency transformation primitives (DFT and FFT butterfly) are also single instructions.

The VSP can perform real 1024 point FFT's in 2.25 ms, complex in 2.35 ms, and can perform logaritms and complex magnitudes in 2.1 and 1.9 usec. By comparison, the T800 takes 100 msec for the complex FFT.

The features and performance of the VSP combine to produce a node architecture that is very efficient for performing standard signal processing operations. By comparison, the T800 is more than an order of magnitude slower. This high speed node alieviates the need to subdivide algoritms like the FFT, thus reducing the required number of processors and decreasing system complexity.

Front End Processor

As the transputer is an entirely digital processor, it has no method for acquiring data from the external world. This data acquisition is the primary function of the Front End Processor. Data is first sampled by a high speed Analog/Digital converter. Two Motorola 56001 Digital Signal Processors then pack the data into data ensembles and scan the data for maximum and minimum stresses. The data is filtered and downsampled to the Nyquist rate for the system. The data is transmitted out an Inmos-compatible serial link for processing in the CADDMAS system.

The functions of the FEP are divided into two separate boards to improve the isolation of the

analog signal from the noisy digital system. The boards are separated using opto-isolators at all interface points.

Software Environment

While hardware can and has been constructed to meet these requirements,the software engineering can be very difficult and expensive. The problem is compounded by the requirements for real-time performance, parallel computation, flexibility, and in situ reconfiguration.

If standard software engineering techniques were used, the problem would be broken into manageable segments and partitioned out over the hardware architecture. Interfaces would be defined between each of these modules and communication protocols would be constructed. At that point, the coding of components and communications routines could commence. The major problems would occur when system integration begins. At that point, any overlooked factors would become apparent. Interfaces would change, possibly causing changes to ripple throughout the entire system. These changes would continue to ripple throughout the system when the actual performance was measured, and when requirements varied slightly. This would be an extremely painful method to attempt the development of such a system. It is doubtful that the development effort would end in success.

Vanderbilt researchers have been working with AEDC for the past five years to develop tools and techniques to manage the software complexity of this large, parallel, real-time instrumentation system. In order to address the problems associated with the development of CADDMAS, I have used *model-based* techniques coupled with a real-time macro-dataflow execution environment. The principal technique is to automatically generate a macro-dataflow computation structure from a declarative model describing the desired system. This new technique, introduced by the Measurement and Computing Systems Group of Vanderbilt, is well suited to a large class of signal processing, instrumentation, and control problems. The primary tools and techniques used here are part of the Multigraph Programming Environment. Using the model-based approach provides several attractive features:

- **Program Synthesis** techniques allow the system complexity to be managed in the model-based format. A simple example of this is a signal processing system. In this case, a model is constructed pictorially which looks like a signal flow diagram. When the system is

ready for execution, that model is analyzed and the corresponding real-time dataflow graph is constructed and started on the execution platform. The algorithm is specified at the highest level. Implementation specific details are handled by the builder. Changes in the underlying hardware architecture or operating systems have no effect on the models. The models are portable across many platforms.

- **Builder/Execution Decoupling** The builder is only loosely coupled to the execution system. None of the signal-flow algorithms interact with the builder, and therefore the builder is not a bottleneck during execution.

- *Scheduling, Synchronization,and Communication* are implicit in the modeling paradigm. The system is synthesized from a library of executable subroutines written in sequential languages.

- *Dynamic Reconfiguration during execution* of a dataflow graph is supported. One scenario is that conditions in the currently executing system reach a point where they require a *structural* adaptation of the system. These conditions re-trigger the *program synthesis*. The builder reconstructs the system to better match the new conditions. Since the executing graph is real-time and continuous, it cannot be stopped so these reconfigurations occur on-the-fly.

The Model-Based Approach to Instrumentation

Development of real-time instrumentation systems requires the crossover of two distinct fields of engineering: signal processing and computer engineering. The signal processing issues are mainly high-level, algorithm dependent issues while the computational complexity of the real-time events and parallel processing synchronization causes numerous low level, computer engineering issues to emerge.

I have combined numeric and symbolic processing to ease the crossover of the two disciplines [1]. The Multigraph Architecture provides a general framework aiding the construction of model-based engineering systems for real-time, parallel computing environments [2]. The main components of the architecture are shown in Figure 2 and described below:

1-9
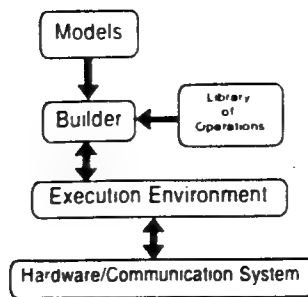
Figure 2: The Multigraph Architecture.

1. The *Symbolic Model Declarations* contain modeling information encoded in a *problem specific* declarative language. For signal processing systems, these declarations are typically of a form that describes the signal flow graph topology as well as specific parameters of the processing blocks in the diagram. On multiprocessor systems, models of the available hardware resources are also included.

2. The *Model Builder* interprets the symbolic model declarations. It creates, initializes, and connects the computation nodes of a macro-dataflow graph. This executable dataflow graph is built as specified by the *Symbolic Model*. In multiprocessor systems, an arc of a graph that crosses processor boundaries automatically sets up inter-processor communication.

3. The *Library of Operations* contains basic sequential programs to be assigned as the computation scripts of the specific nodes of the dataflow graph. For signal processing applications, these are typically numeric routines implementing basic computations (e.g. a FFT or filter). These components are usually implemented in standard numeric languages: C, or Fortran.

4. The *Execution Environment* (also called the Multigraph kernel) provides facilities allowing the model builder to dynamically configure the macro-dataflow graph. As well, the kernel provides a scheduler that controls the graph's execution. The kernels present themselves to the model builder as a single virtual machine even when a set of heterogeneous distributed processing elements are used.

5. The *Hardware and Communication Systems* provide resources to the execution environment needed to run the application as well as a communication path between processors. The Multigraph kernel insulates the user from this level.

<u>Extensions to the Dataflow Model</u>

The macro-dataflow environment supported by the Multigraph kernel has several unique extensions:

1. *Actornodes* are the computational operators of the dataflow graph. An actornode has a state associated with the kernel scheduler. It may be **inactive, active, ready,** or **running**. An **inactive** actor will never be chosen for execution by the scheduler. An **active** actor may be chosen but does not have the proper input control tokens to warrant execution at the preset time. A **ready** actor is waiting in a queue for the processing resources to become available for it to be executed. A **running** actor is currently being executed. Functions to create, destroy, connect input and output ports to datanodes, disconnect from datanodes, activate, deactivated, stop, check the status of, replace, set the context of, set the script of actornodes are all provided for the symbolic interface. An actornode itself is made up from several components:

    (a) The *script* is a piece of code which performs the required operation on data items propagated to the node. It may be coded in almost any language, symbolic or numeric. The script is written in a reentrant fashion so that it can be attached to more than one actor simultaneously. Special Multigraph kernel calls are available to a script programmer for receiving the input control tokens and propagating the resulting token(s). As well, special memory management routines are provided by the kernel for increased performance. Further kernel calls are available for error handling within the kernel.

    (b) The *context* is a static local memory section for a particular actor. It provides two basic services for the actornode. First, it allows the script to have a place to save its state. Thus the actornodes functionality may be changed due to previous values received as input control tokens. Second, it provides an interface to the symbolic builder and control functions such that parameters may be set during the build operation that will affect the operation of the script during execution. For example, a particular signal processing control graph may have an actornode that receives an input vector of time domain data samples, performs a fast fourier transform, and then propagates the result. One implementation of such an actor might be to provide the radix and input data packet size information at build time to the actor through its context. In this case, the

context provides a simple interface for the model builder layer to modify the actor's behavior.

(c) The number of *input* and *output ports* are assigned to an actor when it is created. The ports themselves are the location passed to the kernel routine when an input is to be received or an output is to be propagated. For example, a cross correlation actor may receive two blocks of time domain samples, one from each of the data streams to be correlated. Stream one is connected to one input port and stream two to the other. The result is propagated out the output port. Thus, this actor has two input ports and one output port. An actor with no input ports is always ready for execution.

(d) The *control principle* determines what criteria will be used by the kernel scheduler to decide when an actor's script should be executed. There are currently two choices for this, *ifall* and *ifany*. Under ifall criteria, the actor will only be scheduled for execution after all input ports have data available. The ifany criteria causes the actor's script to be scheduled any time one or more data items are available for input.

2. *Real-time actornodes* are a special variation of actornode. They have two scripts. The first script is executed under the same criteria as a normal actor. Before completing, this first script should make a special kernel call to inform the kernel that the actor is waiting for a particular event to occur (such as an I/O channel communication to complete). The actor will consume no more CPU time until the event occurs. The actor will then be restarted after the event specified, at which time the second script will be executed.

3. *Datanodes* provide a queuing and connection function between actornodes. In order to allow certain portions of the graph to have a higher priority than others, the large-grain dataflow system implemented by the Multigraph kernel is asynchronous. The asynchronous nature of the graph forces the kernel to provide dynamic scheduling functions. As well, a way to queue input tokens between actornodes must be provided. This queuing function is provided by passive kernel components called datanodes. Any number of actor output ports may be connected to a datanode. A datanode may be connected to any number of actor input ports. Datanodes may be enabled or disabled. Unless enabled, a datanode will not allow data to be propagated through it to an actor's input port. Datanodes have been purposely left as a separate component from actors rather than providing the queuing function by way of an actor's input ports. Datanodes provide a simple interface for the symbolic layer to control,

build, and monitor an executing graph. Functions to create, destroy, connect to actor input and output ports, disconnect from actors, clear, read, write, enable, disable, check length, set length, and check connections of datanodes are all provided for the symbolic interface.

4. *Environments* are used to protect system resources and provide a priority mechanism for sections of the dataflow graph. All actors are assigned to an environment. Each environment is given a numeric priority. The kernel scheduler always prefers actors connected to environments of a higher priority. Environments of the same priority are serviced in a round-robin fashion. Only one actor per environment will be executed at any one time. Thus, if two actors need to share a global resource, placing them in the same environment ensures mutual exclusion. Functions to create, destroy, attach to tasks, and set priority of environments are provided for the symbolic interface.

5. *Tasks* provide a generic interface to the basic computational resources of the underlying machine. In a multitasking environment they are simply the different processing threads available to the Multigraph kernel. In a multiprocessor system they are the individual processors themselves. Environments are attached to tasks. The *task* and *environment* concepts allow the actual implementation of the underlying system to be hidden from the user. Thus, the exact same model-based graph building techniques may be used regardless of the underlying hardware architecture. Functions to create and destroy tasks are provided for the symbolic interface.

In summary, the Multigraph kernel provides an interface to build, modify, monitor, and control an asynchronous macro-dataflow graph. The graph may be partitioned by the concept of environments and tasks. The actual dataflow graph looks slightly different from that of a classic dataflow graph due to the introduction of the passive nodes providing queuing functions, datanodes. The computation nodes are called actornodes. They have input ports, output ports, a context, a script, and a status. The use of distinct actornodes and datanodes provides the capability for the symbolic system to build, control, and monitor the dataflow graph.

1-13

Signal Flow Graph Modeling

In order to assist the creation and management of a particular dataflow graph, various tools and languages ranging from basic textual input to a graphical editing environment have been developed [3], [4], [5].

The Multigraph tools HDL and GMB are frequently used to model signal processing systems. HDL stands for Hierarchical Design Language and GMB stands for Graphical Model Builder. The purpose of the language is to represent signal processing systems in a non-algorithmic manner. Rather, the system is described in terms of its *structure*. As such, HDL is classified as a *declarative* language. Block diagrams are often used to represent the signal flow among various modules of a complex signal processing system.

The HDL declarations allow a simple means of describing these blocks and their interconnections. As well, HDL adds an extension to this approach through the concept of hierarchical decomposition. Any one block of the modeled signal flow diagram may itself be represented by a set of blocks (a sub-component, a smaller signal-flow diagram). These hierarchies may proceed to arbitrary levels of abstraction. Complex blocks, made from one or more other blocks, are called *compounds*. Non-decomposable blocks are called *primitives*. The actions to be carried out by primitives are implemented with Multigraph *actornodes*. Each primitive has local parameters that have been inherited/derived through the hierarchy. After design of the top level signal flow graph of the hierarchy is complete, an interpreter (the Model Builder) converts it first to the primitive components of the hierarchy and then maps the result into the parallel execution environment provided by the Multigraph kernel. Each of the nodes of the resulting dataflow graph has a local state derived from the parameters inherited through the hierarchy (the actor's *context*). Node neighbors are known only through the input and output connection arcs. The resulting execution ready program is inherently parallel as a result of the declared signal flow graph being parallel. As well, it has been automatically generated from the high level declarations of the signal processing system and may now be scheduled by the underlying environment. Thus, the computer engineering issues around synchronization have been removed.

GMB is a generic iconic editor program. It runs under X windows and on PC platforms. GMB has been tailored in correspondence with the HDL language. Through this interface, the user may build the declarations of his desired signal flow block diagram in an entirely graphical manner. Figures 3 and 4 show screen dumps from the GMB environment.
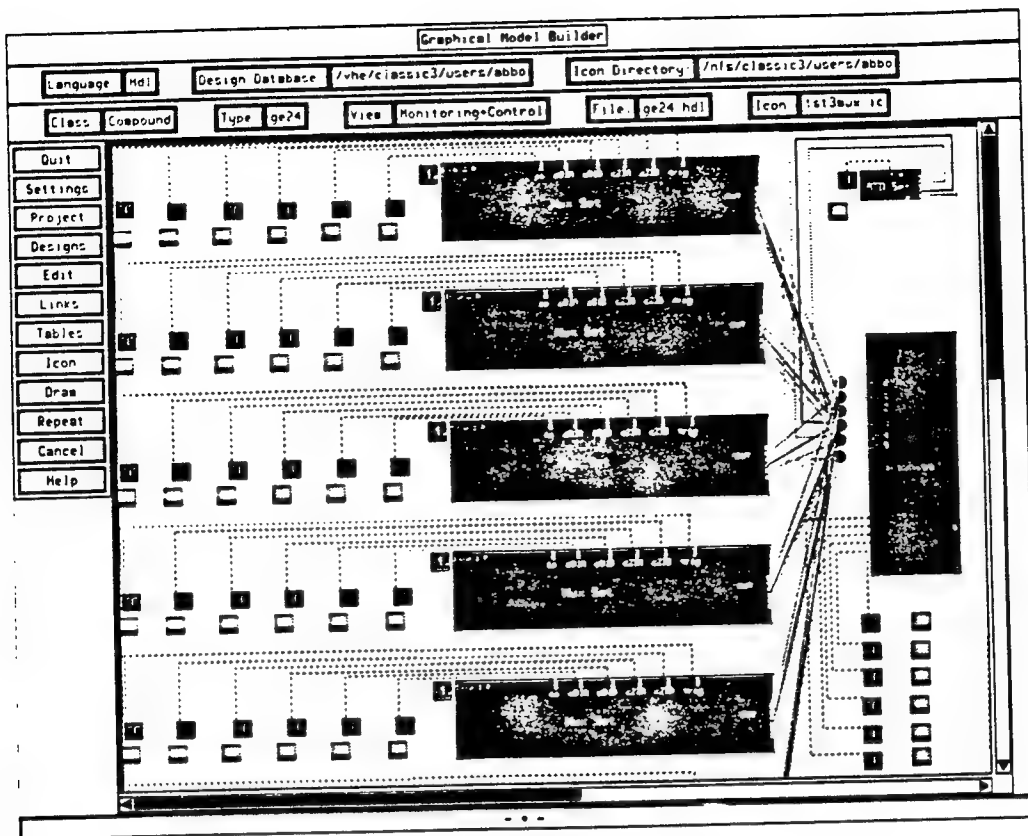
Figure 3: The top level CADDMAS signal flow graph
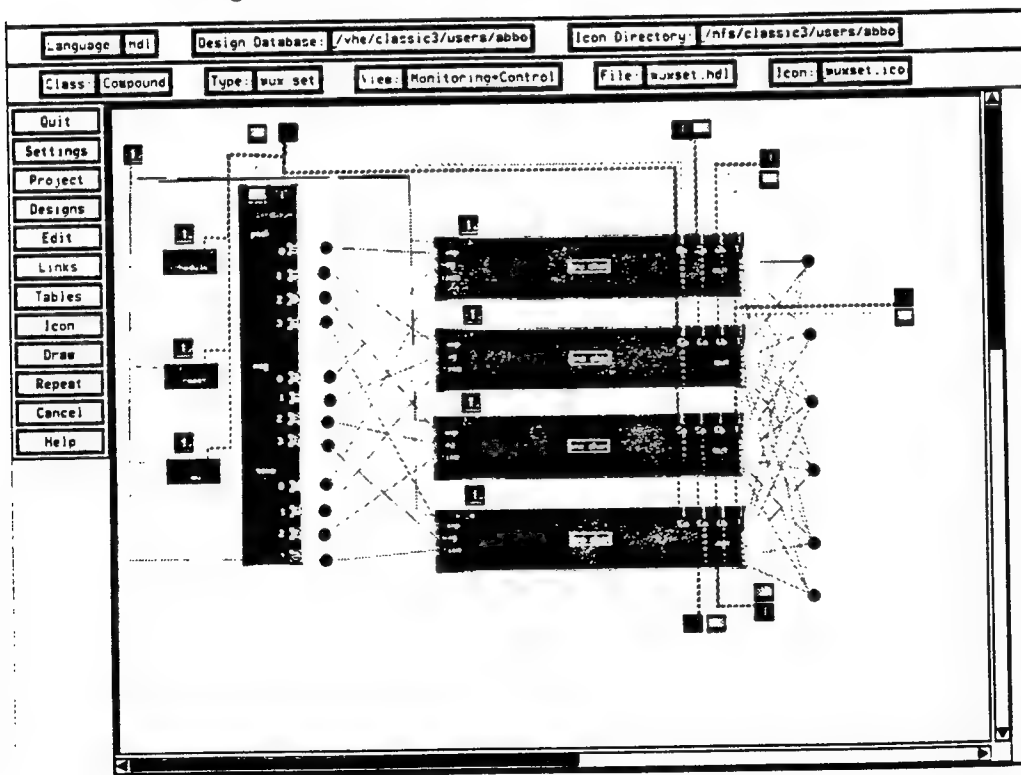
Figure 4: The CADDMAS multiplexor signal flow graph.

1-15

Hardware Architecture Modeling

In order to assist the creation and management of a parallel architecture suitable for a particular application system, the Automatic Parallel Network Analyzer (APNA) tool has been developed. The basic computational blocks of parallel instrumentation systems often allow processor interconnections that are not fixed. For example, transputer architectures with DSP coprocessors fit signal processing applications especially well since the hardware connection architecture can be configured to match the structure of the signal flow graph.

As computational requirements increase, so does the need for more processors. When an architecture contains more than a couple of processors, managing hardware complexity becomes a serious problem. Testing to see if the network of processors has been interconnected correctly, searching for hardware errors, generating network information for an application loader, or creating map files for a message passing system by hand is time-consuming for tens of processors and impossible for hundreds.

Modeling the hardware architecture helps manage this complexity. Our approach again uses the generic graphical model builder GMB. Models of parallel hardware architectures can easily be expressed in graphical terms (i.e. by icons and their interconnections). Hierarchy based modeling is used to manage complexity. The lowest hierarchy levels model the basic processors while the highest level constitutes the complete system. Intermediate levels model boards and subsystems. A number of attributes are attached to the processors (e.g. speed or memory size) in order to capture additional aspects of the system.

The declarative description of the hardware architecture is used to automatically produce load files and maps required by different software components of the system. The APNA tool loads the interpreted models and draws the network in a graphical window. It is capable of loading files of a number of different formats, comparing networks and displaying their differences, generating files for hardware diagnostic programs and a network loader, and displaying the attributes of the processors. It can selectively generate message passing maps, handling multiple paths between processors while optimizing the maps for minimum memory usage. Figures 5 and 6 show screen dumps from the GMB used to model hardware architecture and the APNA tool.
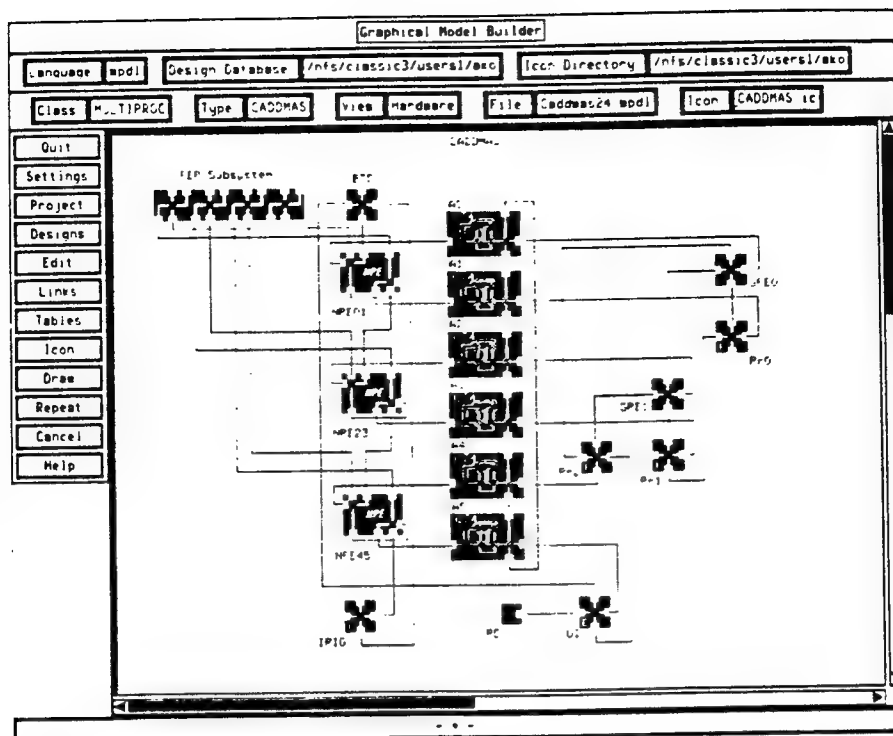
Figure 5: The top level CADDMAS hardware model set up in GMB
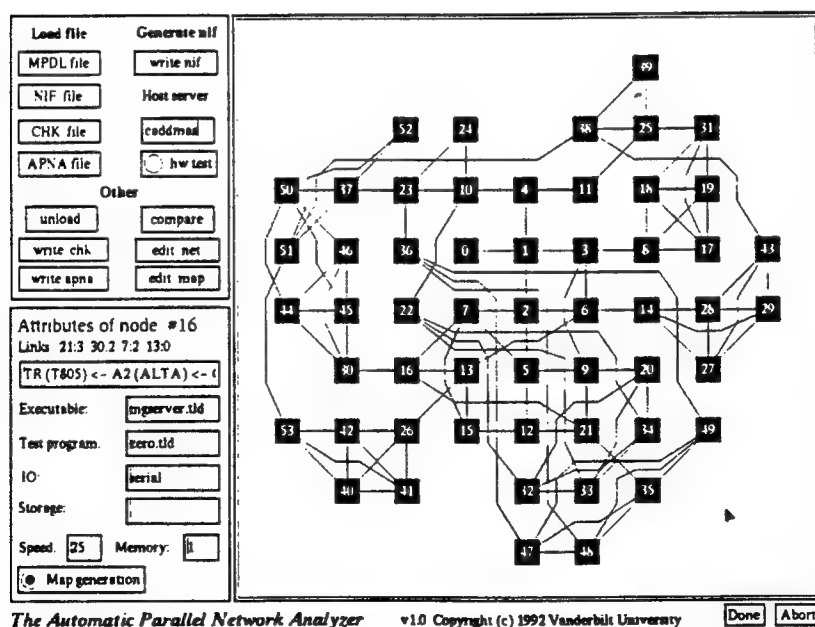
Figure 6: The APNA tool used to analyze the hardware model.

The CADDMAS

The CADDMAS system was developed using the hardware and software methodology previously described. The execution system was synthesized using the models pictured in 3. Each horizontal block represents the processing associated with a single Front End Procesor. The hierarchical decomposition of these blocks is shown in 4. Any changes to the processing performed on all signals is accomplished by modifying the structure of this or lower-level blocks. The hierarchical decomposition allows a top-down approach to these modifications.

To construct a system of arbitrary size, these blocks are replicated in the editor to match the desired number of channels. The blocks are connected to the graphics block to enable display and printing of the processed results.

At the same time, the hardware is modeled in the hardware modeling paradigm. The processor complexes. i.e. physical boards, are hierarically modeled as small blocks. Given these blocks, the overall architecture is modeled. The model interpreter uses these models to construct communications maps, processor load configurations, and test programs for the new architecture.

Another program, the User Interface(UI), performs two critical functions in the CADDMAS system. The graphical user interface allows the user to configure various visualization screens interactively. The user can select the number of visible windows on a screen, the contents of each plot window, and the parameters of each plot, such as titles, labels, axis ranges, and plot type, and display window update rate. Stored configurations automate the operation of the user interface. The user can also print any window or all windows on a screen.

The UI also manages the online reconfiguration of the executing system. As the user request plots, the computation structures must be created from the system models, and unused structures must be destroyed to keep from saturating system memory and processing resources.

Conclusions

The model-based approach has proven to be a very useful tool to help manage the complexity of this large, parallel system. I have now built three different sized CADDMAS systems using the procedure outlined above. The first system was a four channel (single FEP), 8 Campbell, 2 graphics displays system. Actual configuration time for that first system could not be measured

because some of the tools were being finished along the way. The second system was a 24 channel, 48 Campbell, 3 graphics displays, 3 laser printers system. It took one week to physically wire, and get the software running. That time included finding a previously missed bug in the message passing system and making several new patch cables. The third system was a four channel (single FEP), 4 Campbell, 1 graphics display system made from spare parts. It only took four hours to physically wire and get the software running. This proves that the approach is indeed flexible and efficient.

Future plans for the tools and techniques include: better methods to map the signal flow graph to the hardware, provide a dynamic load balance capability, automatically create and destroy pieces of the signal flow graph as required by user interface changes (this addition will greatly help the need for cross correlation operations since pre-building all possible cross correlations would require $(N * (N - 1))$ connections).

Future plans for CADDMAS applications include a 200 channel 50 KHz system using Texas Instruments TI 320C40 processors in conjunction with some of the current transputer technology. The Multigraph, APNA, and HDL tools are sufficiently portable that the added processor type should not be a major problem.

Acknowledgements

References

[1] Sztipanovits, J., and Purves, R., "Coupling Symbolic and Numeric Computations in Distributed Environment", *Proc. of the Workshop on Coupling Symbolic and Numeric Computations*, Seattle, WA. pp.13/0-13/13, 1987.

[2] Biegl, C.: "Design and Implementation of an Execution Environment for Knowledge-Based Systems" Ph.D. Thesis Dept. of Electrical Engineering, Vanderbilt University, Nashville, TN., Dec. 1988.

[3] Karsai, G.: "Hierarchical Description Language (HDL) User's Manual" Dept. of Electrical Engineering, Vanderbilt University, Technical Report #87-004, 1987.

[4] Karsai, G.: "Declarative Programming Techniques for Engineering Problems" Ph.D. thesis Dept. of Electrical Engineering, Vanderbilt University, Nashville, TN., Aug. 1988.

[5] "Research on Intelligent Process Control Systems" Dept. of Electrical Engineering, Vanderbilt University, Technical Report #88-003, 1988.

Prediction of the Performance of a
Proposed Hydrogen Disposition System
Using a Well-Stirred Reactor Model

Julie A. Darnell
Graduate Research Assistant
Department of Mechanical Engineering

University of Tennessee Space Institute
B.H. Goethert Highway
Tullahoma, Tennessee 37388

September 1993

Prediction of the Performance of a
Proposed Hydrogen Disposition System
Using a Well-Stirred Reactor Model

Julie A. Darnell
Graduate Research Assistant
Department of Mechanical Engineering
University of Tennessee Space Institute

Abstract

Arnold Engineering Development Center (AEDC) wants to initiate ground test capability of hydrogen-fueled aeropropulsion systems in the Aeropropulsion Systems Test Facility (ASTF) using a free-jet test configuration. Such testing will potentially produce combustible mixtures of hydrogen and air entering the exhaust ducting, which represents serious safety considerations. Reduction of the level of hydrogen in the exhaust ducting to below flammability limits requires a Hydrogen Disposition System (HDS). AEDC is investigating the technical feasibility of an HDS for ASTF that involves controlled burning of the exhaust gas using piloted, vee-gutter flameholders. Due to deficiencies in the literature and the hydrogen-air flammability database, AEDC has initiated a technology development program to investigate the flammability limits of premixed, turbulent hydrogen-air flows over the proposed HDS hardware for the range of flow conditions in ASTF where flammability is uncertain.

To predict the flame stability performance of the proposed HDS hardware, the author modeled the recirculation zone of the vee-gutter flameholder as a well-stirred reactor. The well-stirred reactor is a one-dimensional model for steady, homogeneous combustion. Application of the conservation equations and the chemical kinetics of hydrogen-air combustion resulted in a set of non-linear, algebraic equations that were solved numerically using the Lewis General Chemical Kinetics and Sensitivity Analysis Code, LSENS. The flame stability limits predicted by LSENS will be compared to DeZubay's experimental database of hydrogen flammability limits and to the HDS technology development database when testing is complete. Results of the comparison of the analytical predictions and the experimental data will be reported when the analysis is concluded.

Prediction of the Performance of a
Proposed Hydrogen Disposition System
Using a Well-Stirred Reactor Model

Julie A. Darnell

Table of Contents

Prediction of the Performance of a
Proposed Hydrogen Disposition System
Using a Well-Stirred Reactor Model

Julie A. Darnell

List of Illustrations

Prediction of the Performance of a
Proposed Hydrogen Disposition System
Using a Well-Stirred Reactor Model

Julie A. Darnell

1.0 Introduction

Arnold Engineering Development Center (AEDC) wants to initiate ground test capability of hydrogen-fueled aeropropulsion systems in the Aeropropulsion Systems Test Facility (ASTF) using a free-jet test configuration. Such testing will produce combustible mixtures of hydrogen and air in the facility exhaust ducting. Reduction of the level of hydrogen in the exhaust ducting to below flammability limits requires a Hydrogen Disposition System (HDS). The HDS will increase the engine operation times over the test envelope before combustible levels of hydrogen are produced in the exhaust ducting. A conceptual design of an HDS for ASTF involves controlled burning of the exhaust gas using an array of piloted, vee-gutter flameholders, similar to those in a gas turbine afterburner. Figure 1 is a simple schematic of the conceptual design for real-time afterburning.
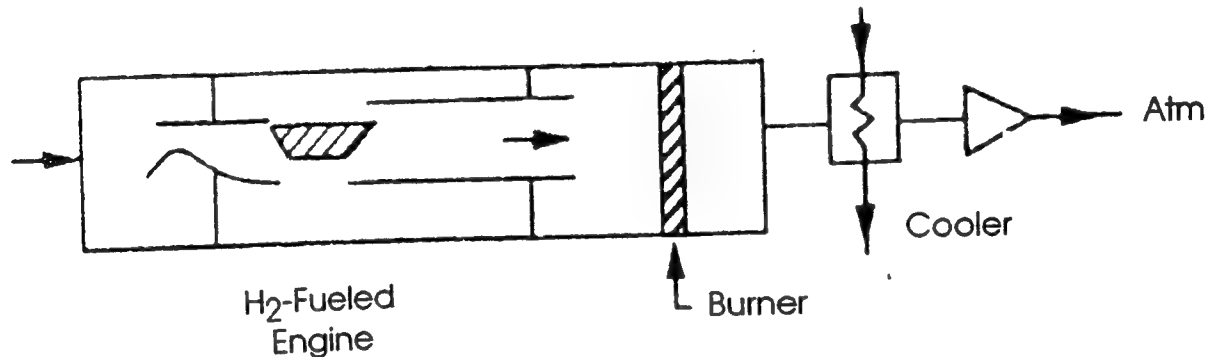
Figure 1: Schematic of the HDS Conceptual Design for Real-Time Afterburning [1]

The facility and capital requirements for this design are feasible; however, technical feasibility of the design over the range of conditions expected in ASTF is uncertain.[1]

Flammability limits of hydrogen-air mixtures are well documented at pressures near atmospheric;[2] however, the literature is deficient concerning hydrogen-air flammability limits on the order of 1.0 psia, the minimum operating pressure of ASTF. Due to these deficiencies,

AEDC is conducting a technology development program that involves a full-scale, section test of the proposed HDS hardware. This investigation will provide a technological database for the actual design of an HDS for ASTF. The database will detail the flame stability limits of premixed, turbulent hydrogen-air flows over the piloted, vee-gutter flameholder for the range of flow conditions in ASTF where flammability is uncertain.[1]

The author investigated the viability of predicting the flame stability performance of the proposed HDS hardware using a well-stirred reactor model. The model treats the recirculation zone of the vee-gutter flameholder as a well-stirred reactor. The governing equations of the model, which incorporate chemical reaction kinetics and the conservation equations, were solved using LSENS, the Lewis General Chemical Kinetics and Sensitivity Analysis Code.[3, 4] When the analysis is concluded, the flame stability limits predicted by LSENS will be compared to predictions based on DeZubay's data [5, 6] and to the experimental results from the HDS technology development test.

2.0 HDS Technology Development Test

AEDC personnel designed the HDS technology development test to investigate the flammability of hydrogen-air mixtures over the range of test conditions expected in ASTF where flammability is uncertain. The test facility, flow simulation requirements and test procedures are described below.

2.1 Test Facility Description

AEDC is conducting the HDS section test in R1A1, a research test cell located in the Engine Test Facility (ETF). Figure 2 is a schematic of the HDS technology development test setup in R1A1. The primary airstream, supplied by atmospheric or high pressure air, flows downstream. The injector rings introduce gaseous hydrogen into the primary flow. The mixing duct provides a homogeneous, turbulent mixture of hydrogen and air at the flameholders. The secondary air supply produces a layer of air around the primary flow to prevent flashback upstream through the boundary layer. The two outer flameholders are centered over the primary and secondary flow and provide the proper boundary conditions for the center flameholder. The center flameholder is immersed in the primary flow and is the focus of study. Hydrogen-air pilot torches situated behind each flameholder provide ignition of the cold flow.

Figure 3 shows a detailed view of the test section. Upstream of the flameholders, the cross section of the primary flow is 4 in x 11 in. The spacing of the
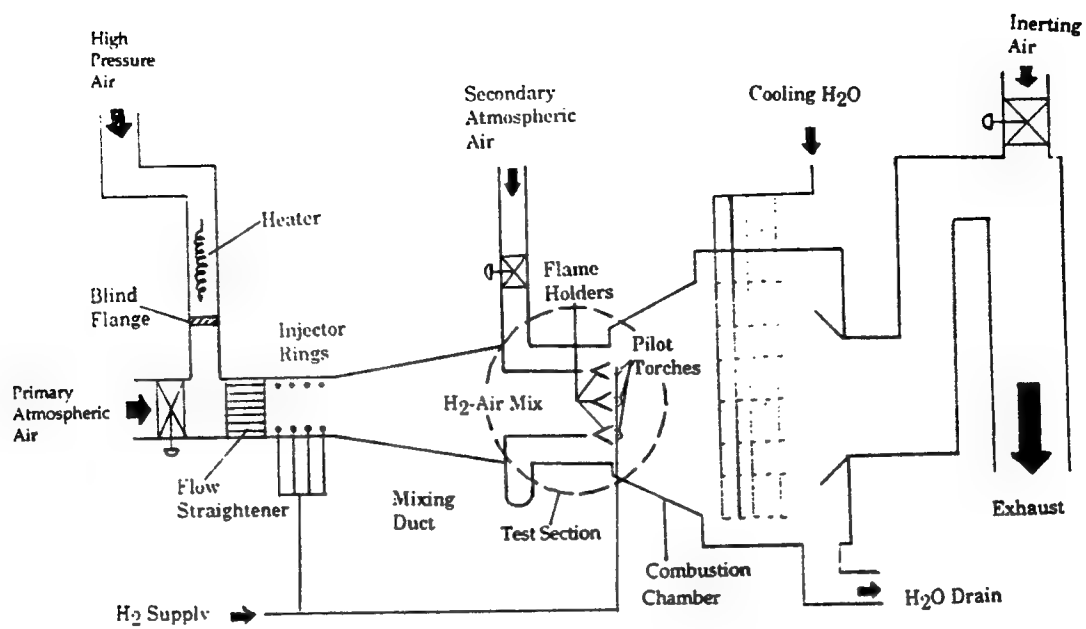
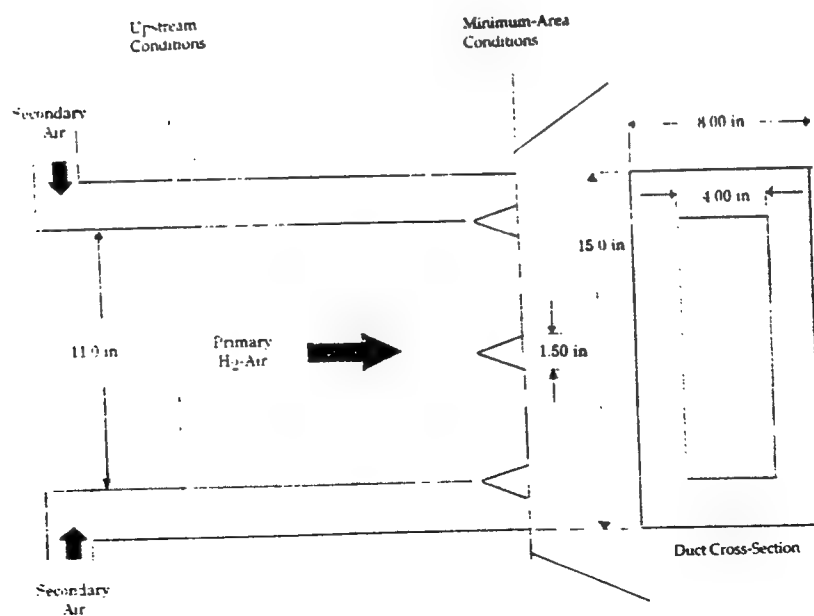Figure 2: HDS Technology Development Test Setup in R1A1

Figure 3: Detailed View of HDS Test Section

flameholders produces thirty percent geometric blockage at the flameholder minimum area. The geometry of the combustion chamber provides a flow separation anchor and approximately constant static pressure immediately downstream of the flameholders. Diagnostic systems include an infrared wavelength radiometer to determine flame light off and blowoff and a white light shadowgraph system for flame visualization.

The flameholder geometry, shown in Figure 4, was designed as a typical vee-gutter flameholder in a gas turbine afterburner.
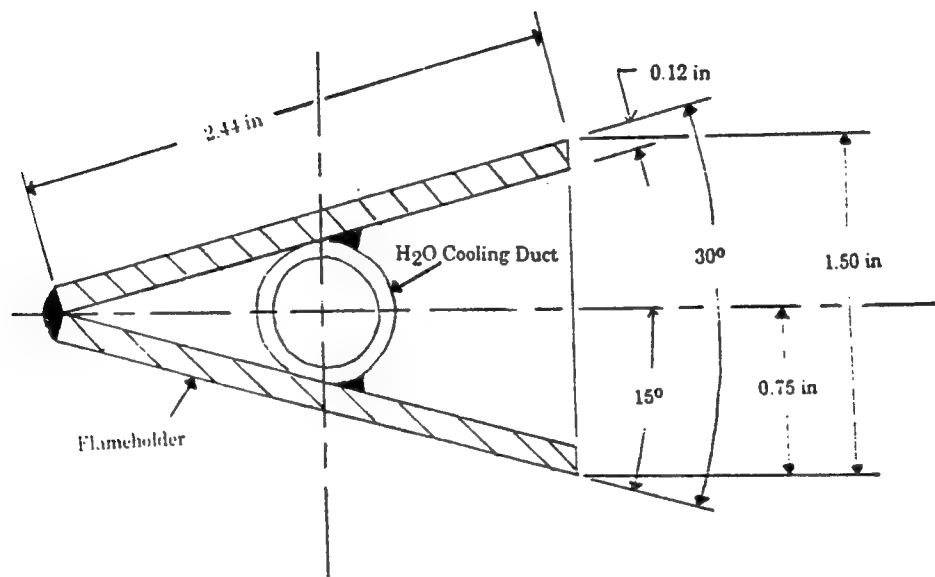
Figure 4: Geometry of the Vee-Gutter Flameholders

The flameholder has a width of 1.5 in with a 30° included angle. The flameholder extends across the duct; however, the length of the flameholder exposed to the primary flow is 4 in. The water cooling duct transfers heat from the flameholder surface.

The exhaust system is designed to cool the hot combustion products, reduce the volumetric flow rate, inert the exhaust gas with air to insure the hydrogen level is below flammability limits, and direct the exhaust gas to the ETF plant exhaust system.[1]

2.2 Flow Simulation Requirements

The HDS section test will investigate the effect of velocity, pressure, temperature and fuel-air equivalence ratio on the flame stability performance of the vee-

gutters. Table 1 outlines the range of experimental flow conditions, defined upstream of the flameholders, for the section test.

Table 1: Experimental Upstream Flow Conditions for HDS Section Test

PARAMETER	RANGE
Approach Mach No.	0.18 – 0.48
Static Pressure (psia)	1.5 – 8.0
F/A Equivalence Ratio	0.2 – 2.0
Total Temperature (^{O}R)	530, 930

The minimum velocity required to prevent flashback upstream and the maximum velocity to produce choked flow at flameholder minimum area bound the approach Mach number range. The minimum static pressure in R1A1 is limited to 1.5 psia, and the maximum static pressure is 8.0 psia because flammability uncertainty occurs at low pressures. The equivalence ratio limits are 0.2 and 2.0. The total temperature of the primary flow is 530 ^{O}R or 930 ^{O}R.[1]

2.3 Test Procedures

The planned test procedures involve defining light off and blowoff limits of lean and rich hydrogen-air mixtures. Figure 5 [1] is a diagram of the test procedure for lean light off and blowoff.
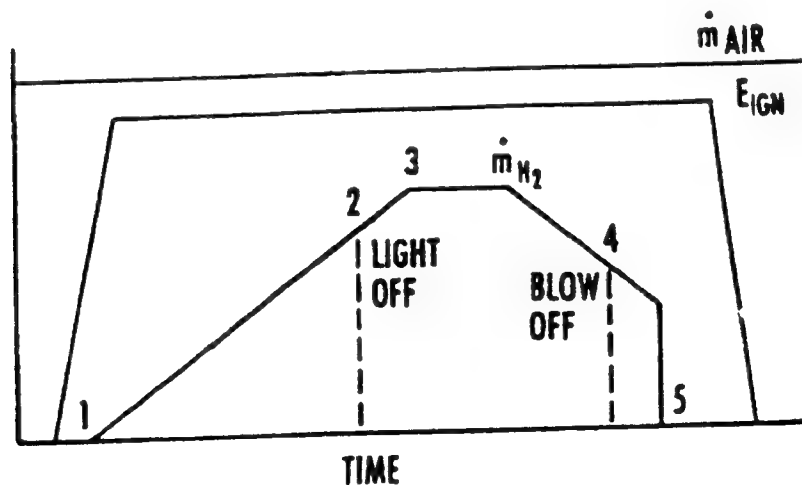
Figure 5: Diagram of Test Procedure for Lean Light Off and Blowoff [1]

Flow conditions for a given air mass flow rate, m_{air}, are set and held constant. Once m_{air} is established, the hydrogen mass flow rate, $mH2$, increases along a linear path. The point of light off (point 2 in Figure 5) is defined, and the flame is stabilized (point 3 in Figure 5) at a higher flow rate for a short time. The point of lean blowoff is defined by decreasing $mH2$ on a linear path until flame extinction occurs.

Figure 6 [1] is a diagram of the test procedure for rich light off and blowoff.
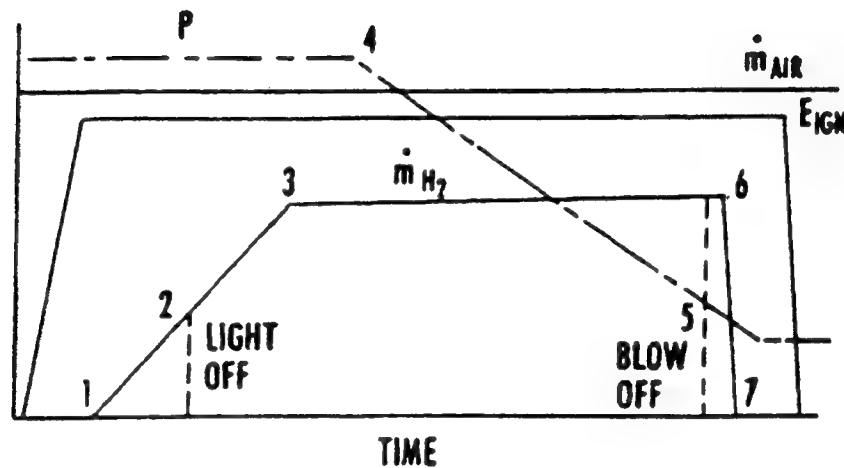
Figure 6: Diagram of Test Procedure for Rich Light Off and Blowoff [1]

The hydrogen-air mixture is ignited following the same procedure outlined for lean light off. Once the flame is stabilized at the desired fuel-air equivalence ratio (point 3 in Figure 6), the point of blowoff (point 5 in Figure 6) is defined by decreasing the pressure, P, on a linear path until flame extinction occurs.

3.0 Theoretical Considerations in Model Development

Before detailing the model used to predict the performance of the proposed HDS hardware, theoretical considerations in the model development are discussed. Theoretical considerations involve: (1) flame stabilization on a bluff body and (2) the well-stirred reactor theory.

3.1 Flame Stabilization on a Bluff Body

Inserting a bluff body into a high-speed gas stream creates a region where the

flow velocity is less than the burning velocity of the mixture. Figure 7 is a schematic of a stabilized flame on a vee-gutter.
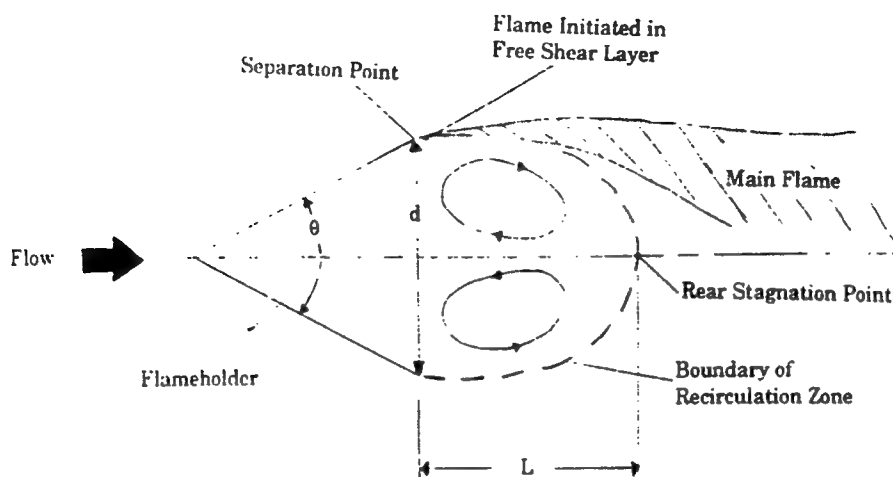

Figure 7: Flame Stabilization on a Vee-Gutter Flameholder

The flameholder produces in its wake a region of recirculating flow. After ignition of the mixture, the recirculation zone acts as a pilot, igniting the fresh mixture in the shear layer and anchoring the flame to the flameholder.

3.1.1 Definition of Flame Stability

Expression of the flame stabilizing performance of a bluff body usually takes the form of: (1) the range of equivalence ratios over which combustion is achieved, and (2) the maximum flow velocity that a system can tolerate before flame extinction.[7] Since the advent of the gas turbine engine, flame stabilization on bluff bodies has been studied extensively.

DeZubay [5, 6] conducted flame stabilization experiments of hydrogen-air mixtures on small, axisymmetrical disks for a range of pressures. DeZubay's study resulted in a correlation between the fuel-air equivalence ratio, the velocity and pressure, defined at the minimum-area. Figure 8 shows DeZubay's flame stability parameter, β, at blowoff as a function of equivalence ratio. Figure 8 also includes a sixth-order polynomial fit of DeZubay's flame stability parameter.
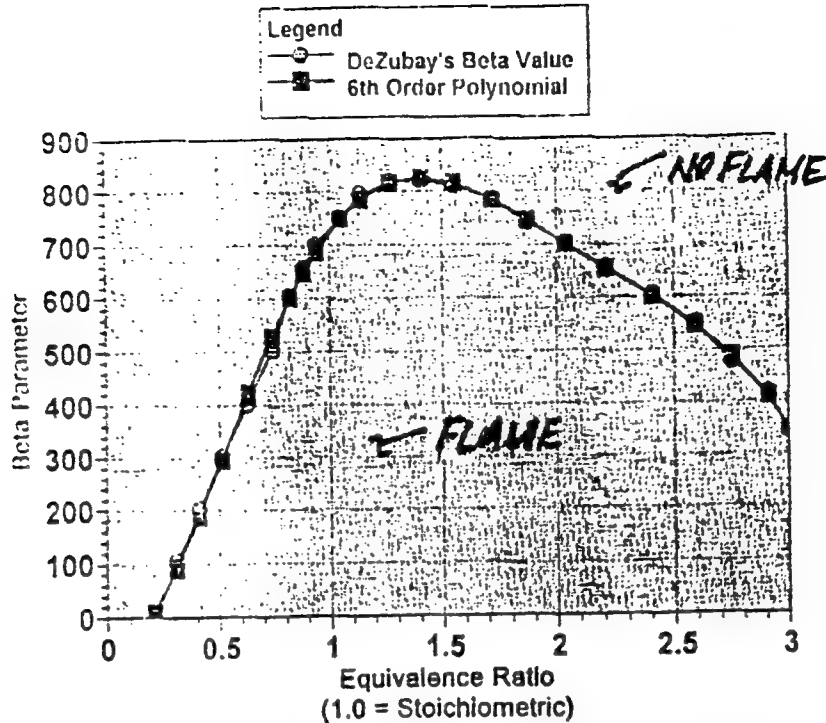
2-11

Figure 8: DeZubay's Flame Stability Parameter for H_2 with a Sixth-Order Polynomial Fit

For hydrogen-air mixtures, DeZubay developed the following correlation:

$$\beta = U / (p^{0.61} d^{0.74})$$

where U is the velocity at the disk (ft/sec), p is the static pressure at the disk (psia), and d is the diameter of the disk (in). For a given velocity, pressure and diameter, β is calculated. If β is less than the blowoff value determined from the equivalence ratio, the flame is stable; if β is greater than or equal to the blowoff value, flame extinction occurs. Zukoski [8] determined that the blowoff velocity for a two-dimensional flameholder is one-half the blowoff velocity of an axisymmetrical flameholder with the same characteristic dimension.

3.1.2 Theoretical Aspects of Flame Stabilization

The theoretical study of the mechanism of flame stabilization on a bluff body takes two basic approaches:[7]

2-12

(1) Following Longwell and associates, [9] the near wake, or recirculation zone, of the flameholder is the focus of study. Analysis of the flame stabilizing mechanism involves viewing the recirculation zone as a homogeneous chemical reactor. Flame blowoff occurs when the residence time of combustion gases in the recirculation zone is less than the time necessary to raise the temperature of the gases to the ignition temperature.

(2) Following Zukoski and Marble, [10] the free shear layer surrounding the recirculation zone is the focus of study. Fresh mixture from the freestream is entrained in the shear layer and ignited by turbulent mixing with previously ignited hot combustion products from the recirculation zone. The burning mixture in the shear layer propagates downstream igniting the neighboring fresh mixture in the freestream. At the end of the wake, some of the burning mixture is entrained into the recirculation zone while the remainder continues downstream. The recirculating flow conveys the burning mixture upstream where it ignites the fresh mixture in the shear layer. This cyclic process is the mechanism of flame stabilization. Flame blowoff occurs when the residence time of fresh mixture in the shear layer is less than the delay time for ignition of the fresh mixture by hot combustion products from the recirculation zone.

On the basis of model simplicity, the author selected the mechanism proposed by Longwell to model flame stabilization on the center vee-gutter. Also on the basis of model simplicity and historical precedence,[11, 12] the author selected the well-stirred reactor theory to model the recirculation zone of the flameholder as a homogeneous chemical reactor.

3.2 Well-Stirred Reactor

The well-stirred reactor, developed by Longwell and Weiss,[13] is a one-dimensional, mathematical model for steady, homogeneous combustion. In the well-stirred reactor, the rate of mixing is assumed to be infinite; therefore, the combustion

gases in the reactor are homogeneous with uniform temperature. Figure 9 shows a conceptual schematic of a well-stirred reactor.
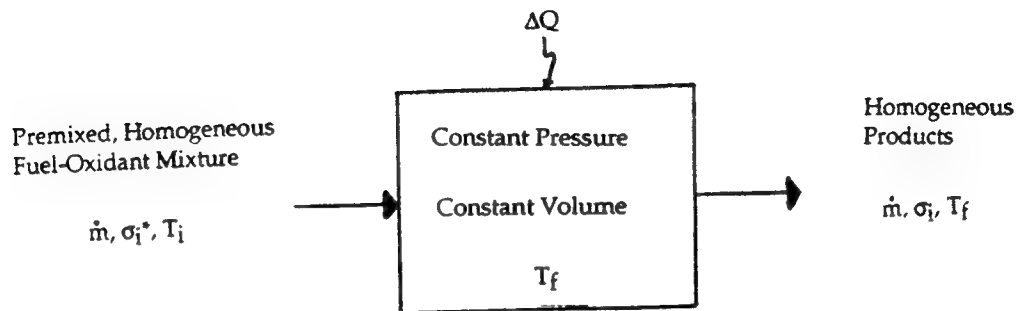
ΔQ

Premixed, Homogeneous
Fuel-Oxidant Mixture

Constant Pressure

Constant Volume

T_f

Homogeneous
Products

$\dot{m}, \sigma_i, T_f$

$\dot{m}, \sigma_i^*, T_i$

Figure 9: Conceptual Schematic of a Well-Stirred Reactor

The premixed, homogeneous reactants enter the reactor volume, V, at a specified mass flow rate, m, and uniform inlet temperature, T_0. The concentration of each reactant species entering the reactor is σ_i^*. The reactor contains combustion gases at a constant pressure, p, and uniform temperature, T_f. Heat, ΔQ, may be added to or extracted from the reactor volume, or the reactor may be adiabatic. For steady operation, the combustion products exit the reactor at a mass flow rate, m, and uniform temperature, T_f. The concentration of each product species leaving the reactor is σ_i.[14]

4.0 Well-Stirred Reactor Model of Proposed HDS Hardware

The recirculation zone of the center flameholder was modeled as a well-stirred reactor. The model assumes the following conditions:

(1) primary flow from upstream to the flameholder minimum area is steady, compressible, inviscid and isentropic;

(2) the flow experiences no body forces;

(3) total pressure and total temperature remain unchanged between upstream and minimum area conditions;

(4) the recirculation zone is an adiabatic, constant pressure reactor,

(5) the hydrogen-air mixture is an ideal gas;

(6) air is standard composition with argon and carbon dioxide inert, and

(7)   the hydrogen injected into the wake by the pilot torches is neglected.

Application of the conservation equations and the chemical kinetics of hydrogen-air combustion results in a set of non-linear, algebraic equations that can be solved numerically.[14] The author selected a general chemical kinetics program entitled the Lewis General Chemical Kinetics and Sensitivity Analysis Code (LSENS) to solve the governing equations of the model. LSENS was developed at the NASA Lewis Research Center.[3, 4]   LSENS requires the following set of input:

(1)   the chemical kinetics of hydrogen-air combustion with Ar and CO_2 inert;

(2)   the fuel-air equivalence ratio of the mixture;

(3)   the initial static pressure and temperature of the mixture;

(4)   the reactor volume (i.e., the recirculation zone volume), and

(5)   the initial and maximum mass flow rates (i.e., mass entrained into the wake) and a mass flow increment.

The volume of the recirculation zone was determined by correlations for vee-gutter flameholders given by Baxter and Lefebvre.[11] The mass entrainment rate into the recirculation zone was determined as a fraction of the mass flow rate through the flameholder projected area at upstream conditions.[12] An entrainment rate of 5% was used for the initial mass flow rate, and a rate of 20% entrainment was used for the maximum mass flow rate. LSENS solves the well-stirred reactor equations for each mass flow iteration and outputs either residence time and outlet conditions (e.g., pressure, temperature, composition, etc.) for a predicted stable flame or an 'apparent blowout' message for a no-flame condition.

5.0 Project Continuation

HDS technology development testing in R1A1 is in progress at AEDC. The author began running the LSENS software for the expected test conditions in R1A1 on August 11, 1993. The flame stability predictions by LSENS is being generated and analyzed. The flame stability limits predicted by LSENS will be compared to DeZubay's database and the HDS technology development database when testing is complete. Results of the comparison of the analytical predictions and the experimental data will be reported when the analysis is concluded. The project will continue through December 1993.

6.0 References

1. G. Garrard and C. Bartlett, "Hydrogen Exhaust Gas Disposition by Afterburning", AIAA-91-5075, AIAA Third International Aerospace Planes Conference, Orlando, FL, December 3-5, 1991.

2. I.L. Drell and F.E. Belles "Survey of Hydrogen Combustion Properties," NACA Report 1383, NASA Lewis Flight Propulsion Laboratory, Cleveland, OH, 1958.

3. D.A. Bittker and V.J. Scullin, "GCKP84 - General Chemical Kinetics Code for Gas-Phase Flow and Batch Processes Including Heat Transfer Effects," NASA Technical Paper 2320, Lewis Research Center, Cleveland, OH, September 1984.

4. K. Radhakrishnan and D.A. Bittker, "LSENS, A General Chemical Kinetics and Sensitivity Analysis Code for Gas-Phase Reactions: User's Guide," NASA Technical Memorandum 105851, Lewis Research Center, Cleveland, OH, January 1993.

5. E.A. DeZubay, "Characteristics of Disk-Controlled Flame," *Aerodigest*, Vol. 61, No. 1, July 1950.

6. E.A. DeZubay, "A Study of Flame Stability Based on Reaction Rate Theory," ASME Paper 54-SA-27, Presented at the Semi-Annual Meeting, Pittsburgh, PA, June 20-24, 1954.

7. K.V.L. Rao and A.H. Lefebvre, "Flame Blowoff Studies Using Large-Scale Flameholders," *Journal of Engineering for Power*, Vol. 104, October 1982, pp. 853-857.

8. E.E. Zukoski, "Afterburners," Chapter 21 of The Aerodynamics of Gas Turbine Engines, edited by G.C. Oates, AFAPL-TR-78-52, July 1958.

9. J.P. Longwell, E.E. Frost and M.A. Weiss, " Flame Stability in Bluff Body Recirculation Zones," *Industrial and Engineering Chemistry*, Vol. 45, No. 8, 1953, pp. 1629-1633.

10. E.E. Zukoski and F.E. Marble, "The Role of Wake Transition in the Process of Flame Stabilization on Bluff Bodies," AGARD Combustion Researches and Reviews, London: Butterworths Scientific Publishers, 1955, pp. 167-180.

11. M.R. Baxter and A.H. Lefebvre, "Weak Extinction Limits of Large Scale Flameholders," ASME-91-GT-234, Presented at the International Gas Turbine and Aeroengine Congress and Exposition, Orlando, FL, June 3-6, 1991.

12. D.W. Roberds, et al, "Measurements of Residence Time, Air Entrainment Rate and Base Pressure in the Near Wake of a Cylindrical Body in Subsonic Flow," AIAA Journal, Vol. 27, No. 11, November 1989, pp.1524-1529.

13. J.P. Longwell and M.A. Weiss, *Industrial and Engineering Chemistry*, Vol. 47, 1955, pp. 1634.

14. A. Jones and A. Prothero, "The Solution of the Steady State Equations for an Adiabatic Stirred Reactor," *Combustion and Flame*, Vol. 2, No. 5, October 1968, pp. 457-464.

SIMULATION TECHNIQUES FOR PRESSURE AND TEMPERATURE DISTORTION

Anthony L. DiPietro Jr.

Ph D Graduate Student

Department of Aerospace Engineering

Virginia Polytechnic Institute & State University

Blacksburg, VA 24060

Final Report for:

Graduate Student Research Program

Arnold Engineering Development Center

July 1993

SIMULATION TECHNIQUES FOR PRESSURE AND TEMPERATURE DISTORTION

Anthony L. DiPietro Jr.
Ph D Graduate Student
Department of Aerospace Engineering
Virginia Polytechnic Institute and State University

Abstract

This is a presentation of simulation techniques utilized to generate pressure and temperature distortion in ground gas turbine engine test facilities. It reveals what is currently available for generating pressure and temperature distortion, ramps, and transients. It also reveals some operational problems, advantages, and disadvantages associated with each simulator. It mentions that as a result from this work, new concepts were conceived for producing pressure and temperature distortion. These concepts were developed to possibly improve upon what was already available in the Arnold Engineering Development Center gas turbine engine test cells and other ground gas turbine engine test facilities. It also mentions that this work is the result of a continuing effort at Virginia Polytechnic Institute and State University to investigate axial flow fan and compressor response to unsteady pressure and temperature distortion and combined pressure and temperature distortion.

SIMULATION TECHNIQUES FOR PRESSURE AND TEMPERATURE DISTORTION

Anthony L. DiPietro Jr.

Introduction

Throughout an aircraft's flight envelope, the inlet and engine must remain aerodynamically compatible. Situations which lead to compressor stall, engine surge, or other malfunctioning of the propulsion system must be avoided or at least reduced to a tolerably low frequency of occurrence. Such situations are produced by a departure of airflow, as delivered from the inlet to the engine, from the ideal of a flow that is uniform in gaseous content, temperature, pressure, and uniformly axial in direction. No real flow ever achieves this ideal however, in the pursuit of high inlet pressure recovery for on design, flows are sufficiently close to it so as to pose no compatibility problems.

Aircraft engine manufacturers are concerned with engine operation under non ideal inlet flow conditions. Specifically, the interest is in engine performance with distorted inlet temperature and pressure profiles. Inlet distortion includes pressure distortion associated with aircraft pitch, roll, and yaw rates and boundary layer separation from the fuselage and inlet duct. Temperature distortion is produced by the ingestion of exhaust gases from armament firings such as guns, rockets, and missiles. Also, gas turbine engine exhaust gas re ingestion that occurs in helicopter and VTOL/STOL aircraft operation. Aircraft engine manufacturers as well as engine test facilities are making an effort to develop devices that can simulate these pressure and temperature distortions for direct connect tests on gas turbine engines. These devices produce actual flow conditions at the engine inlet that may exist during an aircraft maneuver or armament firing. Data regarding engine performance and stability under distorted inlet flow conditions, generated by these devices, can now be obtained. This information will be useful in making remedial measures to the propulsion system or to redesign the propulsion system to tolerate the distortion.

Methodology

The method of generating a pressure distortion at the face of a gas turbine engine inlet is to produce a physical blockage in a sector of the inlet. This physical blockage locally reduces the axial velocity of the airstream directly downstream from the blockage. This reduction in axial velocity translates into a local reduction in total pressure behind the blockage. This physical blockage may be a screen, a mechanical vane, or some other physical obstruction. Another method of generating pressure distortion is to mount a series of airjets in the inlet duct near the engine face. These jets are mounted such that the nozzle exits of all the jets are in the same crossectional plane and point directly into the inlet flow. These airjets discharge high pressure compressed air into the oncoming inlet flow at high velocities. A momentum exchange occurs between the locally injected high velocity compressed air from the airjet nozzles and the oncoming inlet flow to produce a local reduction in fluid axial velocity. This reduction in fluid axial velocity translates into a reduction in total pressure. The method of generating a temperature distortion at the engine face is to mount some type of heater or burner in the inlet duct . This heater or burner may be subdivided into sectors that operate independently of each other to locally increase the temperature in sectors of the inlet duct. Another method of generating temperature distortion at an engine face is to actually mount and burn a missile or rocket motor in front of the inlet. It is desirable that the pressure and temperature distortion devices have the ability to change patterns while a test is running. This ability allows transients to be simulated at the engine inlet that may exist during an aircraft maneuver or armament firing.

Results

The Naval Air Propulsion Test Center (NAPTC) utilized a distortion valve to produce steady state and transient pressure distortion. The distortion valve consisted of a square frame with a series of parallel flat vanes. These vanes were connected to operate mechanically as a "venetian blind" to block

quadrants of the inlet. The valve was subdivided into four independently controlled square quadrants. The distortion valve produced good spatial pressure distortion however, it could not produce complex distortion patterns. Another disadvantage to the distortion valve is that the crossection of the inlet had to change from square to circular for direct connect tests. This transition would alter the pattern as it left the distortion valve.

The National Aeronautics and Space Administration (NASA) Lewis Research Center utilized an airjet distortion generator to produce transient pressure distortion. it consisted of 54 airjets mounted in the inlet duct of an engine test cell. Six main control valves mounted outside the inlet duct controlled the discharge of compressed air to the jets. Each valve was installed at 60 degree intervals around the inlet duct and supplied air for nine jets. The generator operated by producing a momentum exchange between the injected compressed air and the inlet duct airflow to obtain a reduction in total pressure. The device produced good controlled spatial pressure distortion and could produce transients during a test. However, it had a complex valve, air transport, air conditioning, and control system. During tests, the system experienced problems with the control valves. NASA also utilized a rotating screen assembly to generate steady state pressure distortion. It consisted of a mount that could support precut screens of varying solidity. This mount could rotate ± 30 degrees and was anchored in the inlet duct of an engine test cell. The device produced good spatial pressure distortion and was relatively simple and inexpensive. A disadvantage with the screen assembly was it could not produce transient while a test was running. It was limiting to the types of patterns that could be produced. It also had a problem with the screens breaking apart during operating conditions with high mass flow.

The Arnold Engineering Development Center (AEDC) also developed an airjet distortion generator to generate pressure distortion for engine tests. It consisted of 56 air injector ports that were all independently controlled and mounted in radial, aerodynamically, streamlined, supporting struts. This strut assembly was anchored in the inlet duct of an engine test cell with the airjet nozzle exits pointing into the oncoming inlet airflow. Utilizing a momentum exchange between the primary air in the inlet duct and secondary air injected by the nozzles, the generator produced controlled spatial pressure distortion

and pressure transients during a test. It had the ability to produce more complex patterns than the NASA airjet because each injector port was independently controlled. It had a complex valve and control system and possessed some of the same disadvantages as the NASA airjet. However, it is speculated that the AEDC airjet generates less freestream turbulence than the NASA airjet because the airjets are mounted in aerodynamically streamlined struts.

A pressure distortion device developed by General Electric in Cincinnati Ohio was their P3G which is a planar pressure pulse discrete frequency generator. It consisted of a rotor driven by an electric motor between two stators. It was a relatively simple device and produce a good planar pressure pulse. The frequency of the pulse was controlled by the rotor speed and the amplitude of the pulse was controlled by the spacing between rotor and the stators. It was limiting because it could only produce one particular type of distortion pattern, a planar pressure pulse, and could not produce transients.

The NAPTC had two techniques for producing temperature distortion. One was a boiler plate rocket motor gas generator mounted directly in front of the inlet duct of an engine. It was designed to burn specially fabricated grains of solid rocket propellant. The other simulator utilized a cylinder that burned solid propellant inside two separate powder chambers outside the inlet duct. A circular manifold channeled the combustion gases around the inlet duct and injected it into the inlet flow through the inlet duct wall. Both simulation devices produced correct gas chemical composition and temperature ramps. Peak temperature ramps up to 10,000 F/s were achieved. An advantage to the rocket motor that burned propellant outside the inlet duct and channeled the combustion gases into the inlet was that it had two powder chambers. These chambers could be ignited at different times to simulate multiple rocket firings. Both simulation devices had the disadvantage of being unable to produce controlled spatial temperature distortion. Other disadvantages are that the powder chambers had to be refilled after each firing and that corrosive agents HCL and Al2O3 solid particulates were produced and attached to engine internal component.

The NASA Lewis Research Center designed, fabricated, and operated two gaseous fuel hydrogen burners to simulate temperature distortion. One thermal distortion generator was designed for large

3-6

engines such as turbojets and turbofans. The other was a smaller thermal distortion generator for use on turboshaft engines. The burner used for the turbojet and turbofan engines was a circular, four quadrant, segmented, hydrogen fueled burner. It utilized fuel injectors and V-gutter flameholders with swirl can ignitors. Each quadrant was independently controlled to produce steady state or transient temperature distortion. The turboshaft engine thermal distortion generator was circular and subdivided into eight 45 degree sectors. It had three swirl cup combustors in each sector for a total of 24 swirl cups. Both burners produced good spatial temperature distortion and temperature ramps. Temperature ramps up to 17,750 R/s were achieved with the large distortion generator. Both burners produced incorrect gas chemical composition due to the hydrogen and air combustion reaction. Hydrogen was used as the fuel because of its low flammability limit, fast ignition, high temperature rise, and steep temperature ramps. A problem with both designs was uniform ignition in each sector/quadrant. Experimental tests on a turbofan engine indicated that the hottest most rapidly changing temperatures occurred at the center of the burner and produced core compressor stalls but not fan stalls.

AEDC also developed two thermal distortion simulation techniques. One simulator was a modification to their airjet distortion generator. Instead of injecting ambient temperature compressed air, the air was heated. The injection of high temperature compressed air produced the thermal distortion. This technique was easily accomplished with minor hardware modifications to the airjet system. It produced good spatial temperature distortion however, pressure distortion accompanied the temperature distortion. It had the advantage of not having to stabilize a combustion reaction inside the inlet duct where the high velocity air was flowing. Some disadvantages were the airjet produced low temperature ramps and required large volumes of hot air to achieve a desired temperature rise. The other temperature distortion simulation device at AEDC was a turnstile mount that supported four small solid rocket motors all at 90 degrees to each other. This turnstile mount had the capability of being positioned at several distances from the inlet. This simulator produced good temperature ramps and correct gas chemical composition. It did have some of the same disadvantages as the rocket gas generators used at the NAPTC. Specifically, it produced corrosive agents HCL and Al2O3 solid particulates that attached to and damaged

engine internal components.

An effort to investigate fan and compressor response to unsteady pressure and temperature distortion and combined pressure and temperature distortion was begun at Virginia Polytechnic Institute and State University in their Aerospace and Mechanical Engineering departments. The effort initiated with the design, fabrication, and operation of a gaseous fuel burner to simulate unsteady temperature distortions associated with gas turbine or rocket engine exhaust gas ingestion into axial flow fans and compressors. The result was a cylindrical four quadrant propane burner that led to a Masters thesis. The device consisted of an 18 inch diameter cylinder, three feet in length, that was subdivided into four quadrants by use of steel plates. Occupying each quadrant was a centerbody that contained a fuel injector and ignition plug. The device was specifically designed to produce temperature transients and unsteady temperature distortion with multiple quadrant combination patterns. However, it did have the ability to produce steady state spatial temperature distortion. The device produced good spatial temperature distortions however the temperature ramps produced were low. It was speculated that if gaseous hydrogen was used as the fuel, steeper ramps would be generated. An effort is currently underway to develop a similar device that will generate an unsteady pressure distortion and the goal is to use the two devices in combination on a multi-stage transonic fan. This work will lead to a Doctoral dissertation at VPI.

Conclusions

Two techniques exist for producing a temperature distortion at the face of a gas turbine engine. One technique utilizes an actual rocket motor that is mounted and burned at the engine inlet. The other technique involves the mounting of a burner or heater in the inlet duct that is capable of locally raising the temperature in sectors of the inlet. Several devices exist for producing a pressure distortion at the face of a gas turbine engine. One class of these devices operates by producing a physical obstruction in a sector of the inlet. The velocity downstream from this obstruction is reduced and this produces a total pressure drop behind the obstruction. The other class of pressure distortion devices operates by utilizing an axial

3-8

momentum exchange between the primary inlet duct airflow and secondary air that is injected directly in the counter direction of the primary airflow. Again, a velocity reduction in a sector of the inlet duct is obtained by a reduction in airflow axial momentum of the primary airstream in that particular sector. This translates into a total pressure drop in that particular sector.

A large amount of published data, engine test data, demonstrates that pure temperature ramps at the engine face with rates on the order of 2000-10,000 R/s can induce engine surge or combustor blow out. Tests indicated that the magnitude of the temperature ramp had a more pronounced effect than either simulated steady state spatial temperature distortion or the absolute level of temperature obtained during the transient.

A plot of inlet total pressure amplitude verses pressure pulse duration data obtained by the NASA Lewis Research Center airjet system indicated that the longer the pulse, the lower the total pressure. Stall occurred during long pulses only which indicates that stall is likely to occur over low frequency ranges. A possible explanation is that there is a finite time interval required for a rotor blade to stall in response to a total pressure distortion. During shorter durations, or higher frequencies, the distortion is too brief for the rotor blade to stall. Results indicate that the stall tolerance is a function of not only the instantaneous distortion level, but also the rate of change of inlet pressure and dwelling time of the fan or compressor rotor blading in an engine inlet distortion.

This work indicates that there are several devices available for producing pressure and temperature distortion. The review also revealed that each device has certain operational problems, advantages, and disadvantages associated with its design. The type of distortion research and experiments that will be conducted will dictate which simulation device is appropriate for that particular application. This work is a continuation of the effort being conducted at Virginia Polytechnic Institute and State University to investigate fan and compressor response to unsteady pressure and temperature distortion and combined pressure and temperature distortion. This work is also a contribution to the effort currently underway at the Arnold Engineering Development Center to establish the capability of pressure and

3-9

temperature distortion testing in the AEDC gas turbine engine test cells. Resulting from this work came new concepts for generating pressure and temperature distortion that may be better tailored for the AEDC gas turbine engine test cells and VPI fan and compressor rig research. Some of these concepts may be an improvement of what is already available at AEDC and other ground gas turbine engine test facilities. These concepts may be investigated further at AEDC and VPI.

REFERENCES

1) Baumbick, Robert J., " Device For Producing Dynamic Distortion Patterns At Inlets Of Air-Breathing Engines", NASA-TM X-2026, Lewis Research Center, Cleveland, OH 44135, June 1970.

2) Hubble, J.D. and Smith, R.E., " Evaluation Of An Airjet Distortion Generator Used To Produce Steady- State Total Pressure Distortion At The Inlet Of A General Electric F101-GE-100 Turbofan Engine", AEDC-TR-78-73, Arnold Engineering Development Center, TN 37389, August 1979.

3) Biesiadny, Thomas J., Braithwaite, Willis M., Soeder, Ronald H., and Abdelwahab Mahmood, " Summary Of Investigations Of Engine Response To Distorted Inlet Conditions", NASA-TM-87317, Lewis Research Center, Cleveland, OH 44135, September 1986.

4) Anderson, R.E., " Aircraft Engine Inlet Pressure Distortion Testing In A Ground Test Facility", AIAA-83-1233, Naval Air Propulsion Test Center, Trenton, NJ, June 1983.

5) Pawlik, Eugene V. and Jones, Robert E., " Experimental Evaluation Of Swirl Can Elements For A Propane Fuel Combustor", NASA MEMO 5-15-59E, Lewis Research Center, Cleveland, OH 44135, June 1959.

6) Rich, William A., " The Simulation Of The Ingestion Of Missile Exhaust By Turbojets", Naval Air Propulsion Test Center, Trenton, NJ.

7) Kutschenreuter Jr., P.H., Collins, T.P., and Vier III, W.F., " The P3G- A New Dynamic Distortion Generator", AIAA 73-1317, General Electric Company, Cincinnati, OH, November 1973.

8) McIlveen, M.W., " Further Test Results With The Airjet Distortion Generator System", AIAA 79-1185, Arnold Engineering Development Center, TN 37389, June 1979.

9) Overall, B. W. and Harper, R.E., " The Airjet Distortion Generator System: A New Tool For Aircraft Turbine Engine Testing", AIAA 77-993, Arnold Engineering Development Center, TN 37389, July 1977.

10) DiPietro Jr., Anthony L., " Design And Experimental Evaluation Of A Dynamic Thermal Distortion Generator For Turbomachinery Research", Masters Thesis, Department Of Aerospace And Ocean Engineering, Virginia Polytechnic Institute And State University, Blacksburg, VA 24060, April 1993.

Design of an Aluminum Combustion Simulator

Robert J. Geierman
Graduate Research Assistant
Mechanical Engineering

University of Tennessee Space Institute
B. H. Goethert Parkway, MS 01
Tullahoma, Tennessee 37388

Design of an Aluminum Combustion Simulator

Robert J. Geierman
Graduate Research Assistant
Mechanical Engineering
University of Tennessee Space Institute

Abstract

This research presents the design of a combustion device which will simulate the aspects of aluminum particle combustion inside a solid rocket motor. At the same time this combustor will allow optical data to be collected during the combustion process.

The first step in this combustor design was to examine past experiments and methods which have been used to examine the combustion of aluminum particles. Next, the fuels and oxidizers, which will be used in the first planned combustion experiments, were selected. The flame temperatures and transport properties of these fuels and oxidizers were calculated using a version of a NASA chemical equilibrium code. The third step in this combustor design was to define the physical dimensions and maximum operating times which would be allowed. These parameters were defined through simplified stress analysis and heat transfer calculations, respectively. The final steps in this combustor design process were to define the procedures and instrumentation to be used during the operation of this combustor.

Design of an Aluminum Combustion Simulator

Robert J. Geierman

Introduction

For many years, metals have been used as additives in solid rocket motor propellants. The reasons for the use of these additives include their ability to help eliminate combustion instabilities and their ability to increase specific impulse. By far, the most popular metal additive is aluminum. The popularity of aluminum usage is due to its high heat of formation (of its oxides) and its relatively low cost.

Although aluminum is widely used as a propellant additive, much of its combustion characteristics and optical properties, especially during solid propellant burning, have not been quantified. By obtaining a better understanding of aluminum combustion, one could possibly increase the specific impulse of solid rocket motors without degrading combustion stability. The increase in specific impulse could be accomplished by either increasing the aluminum content to near its stoichiometric limit, while maintaining a high combustion efficiency, or by minimizing the two-phase flow losses. Similarly, an adequate understanding of the optical properties of aluminum is important to predict the radiation heat transfer to the propellant surface and the spacecraft itself, from the exhaust gases. In order to gain a better understanding of the combustion of aluminum and its properties, it becomes necessary to observe the process in the environment which is found inside a solid rocket motor. Since it is nearly impossible to observe the phenomena inside an actual solid rocket motor, a device is necessary which closely simulates these conditions while at the same time allows for observations of exhaust or combustion process to be made. This report contains an overview of the design process for a combustor which will simulate the combustion of aluminum under solid rocket motor conditions. For a more detailed report see [1].

Methodology

In order to design this combustor, it was necessary to first examine the past experiments which have investigated the aluminum combustion phenomenon. From an extensive examination of many reports, it was found that there is still a lack of information dealing with the aluminum combustion process. Never the less, the previous experiments were divided into three groups. These groups consisted of burning of propellant samples, particle/flame burners, and exotic methods. Burning propellant samples was accomplished by simply igniting propellant samples in some sort of combustion bomb and observing the process (mainly through the use of high speed cameras). Particle/flame burners merely consisted of injecting aluminum particles into a flame or torch. The drawback of these experiments was due to the combustion process in the flames occurred at atmospheric pressure rather than the high pressures found in typical solid rocket motors. The exotic methods consisted of the use of shock tubes, ignition of large particles with lasers, and electrical ignition of aluminum wires. These exotic methods usually burned aluminum in shapes and sizes which were not typically found in solid rocket motors.

All of these experiments observed the combustion of aluminum in one way or another. However, these experiments often failed to burn the aluminum under conditions which are found in solid rocket motors. Additionally, these experiments rarely collected any optical data from the burning aluminum and aluminum oxide particles. For these reasons, it was decided to design a combustor which would allow for observations of aluminum combustion under a simulated solid rocket motor environment. This combustor should also be versatile enough to observe the effects of individual species on the aluminum combustion process. This combustor concept should be inexpensive to build and operate. For these reasons, it was decided to start with a combustor design which would be as simple as possible. The original combustor should start as simple as possible and gradually, as more knowledge is gained, work towards more complex experiments. Keeping this in mind, the original combustor was designed to use a coaxial injector with the aluminum particles injected with the gaseous fuels (see Figure 1). This original combustor concept was
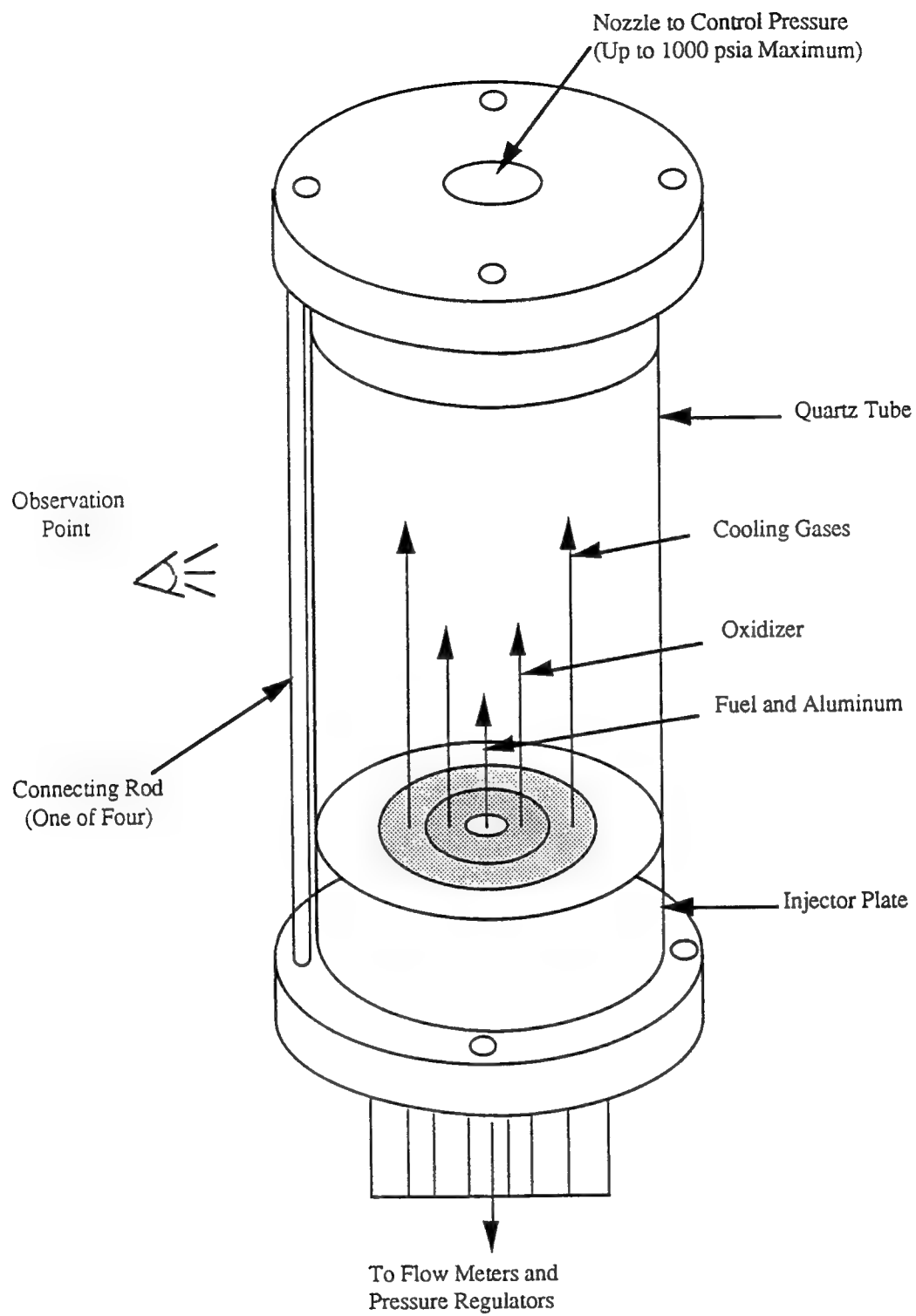
Figure 1: Solid Rocket Motor Aluminum Combustor Simulator
(Simplified Schematic)

later modified. The reasons for these modifications are discussed in the conclusions and recommendations section.

Since the goal of the original combustor concept was to be as simple as possible, it was decided to use both gaseous fuels and oxidizers in the original design (and small mass fractions of aluminum particles (0.1 to 5 percent)). Because hydrogen and water vapor are prevalent products of combustion in solid rocket motors, it was decided to use gaseous hydrogen as the fuel to be used in the first combustion experiments. It was also decided to use methane as the second fuel in this combustor. In this way the effects of carbon monoxide and carbon dioxide can be studied along with the effects of hydrogen and water vapor. Eventually, this combustor will be capable of simulating the majority of species produced in a solid rocket motor.

Before the specific design characteristics of this combustor could be determined, it was necessary to perform a combustion analysis of the fuels to be used. This combustion analysis was performed by using the NASA equilibrium code (SP-273 [2]) to determine the chemical species which will be present in this combustor. It was found from these species that a significant mole fraction of aluminum hydroxide formed during the combustion of low mass fractions of aluminum. As the aluminum mass fraction was increased to beyond 1 percent of the total fuel and oxidizers, the mole fraction of aluminum oxide produced was greater than the mole fraction of aluminum hydroxide. It was therefore recommended that the combustor be operated at aluminum mass fractions greater than 1 percent of the total flow. In addition to the chemical species present, this code also predicted the flame temperature and transport properties of the combustion products. These properties were then used to predict the heat transfer rates and physical size of this combustor.

After the combustion characteristics were calculated, it was possible to determine the physical dimensions of the combustor. Since this initial combustor design should be as inexpensive as possible, it was deemed necessary to make this combustor as small as possible. This not only makes the design less

costly (due to the use of less material), but it also makes for a safer design. A smaller volume at high pressure should cause less damage, than a large volume, in the event of failure. In any event, by starting with a small inexpensive combustor concept, it should be possible to scale to larger combustors if desired later in this research program.

Since this initial combustor will be as small as possible, it was felt that a central fuel injector with a 0.25 in. outside diameter would be sufficient for the first two fuels to be used. By neglecting the two-phase flow effects of the aluminum particles (due to their low mass fraction of the total flow), it was possible to use the O/F ratios (corresponding to the maximum flow rates) to size the other injector ports. Again, to minimize the cost of this combustor, it was decided to use standard size stainless steel tubing for the injectors. The lengths of these injector tubes were designed to create a constant velocity at any radial location. Also, the thickness of the injection tubes was found using a simplified pressure vessel equation and a factor of safety of 2. The dimensions and layout of the injector face are shown in Figures 2 and 3, respectively.

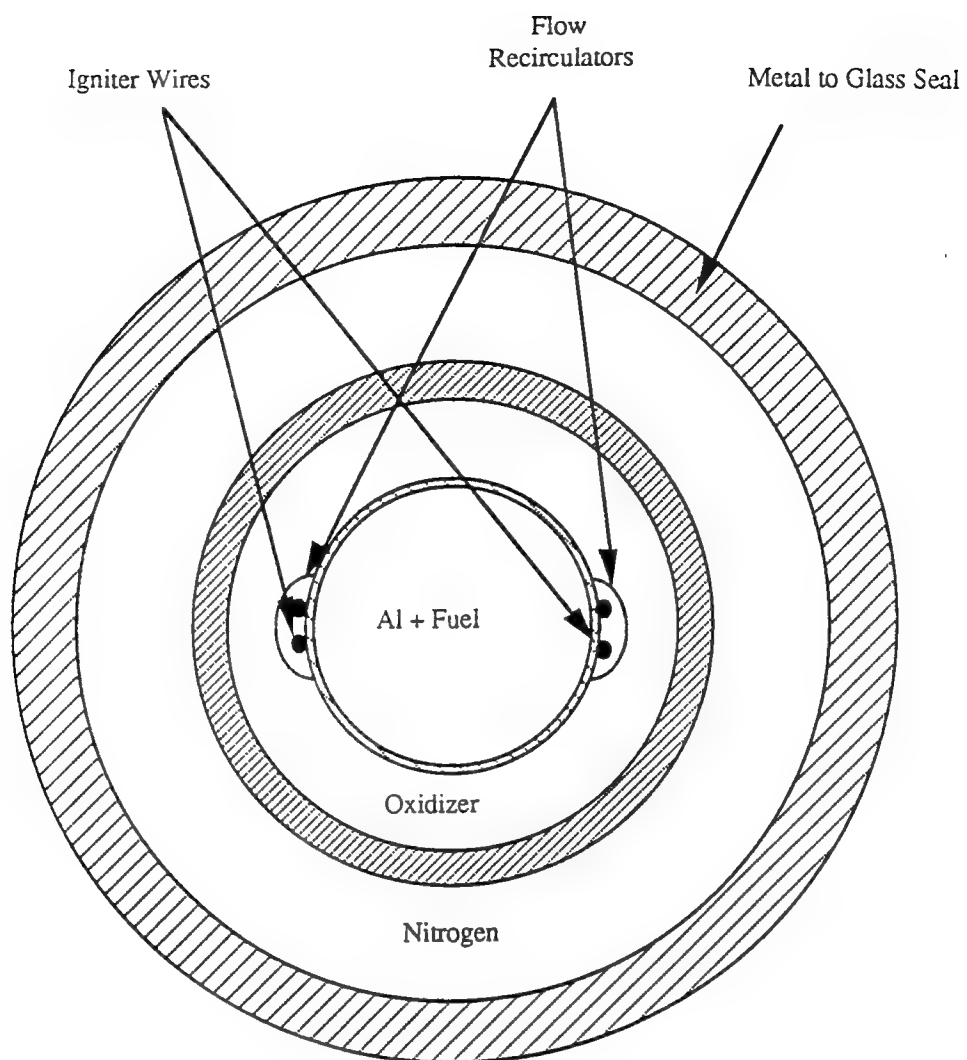| Injection Tube	Inside Diameter (in.)	Outside Diameter (in.)	Tube Length (in.)
Fuel	0.238	0.25	4.5
Oxygen	0.710	0.75	4.75
Nitrogen	0.930	1.0	4.75

Figure 2: Injector Dimensions

Figure 3: Injector Face Layout

The next step in this combustor design was to determine the dimensions of the nozzle. In order to minimize the cost of this combustor, the original combustor concept was designed with a variable area throat (see Figure 4). The nozzle was to be cooled with water and the throat area was chosen to provide for choked flow at the maximum expected flow rates. Heat transfer calculations were performed to determine the necessary flow rates for the coolant water. The dimensions of the nozzle were determined with the same pressure vessel equation used for the injector calculations. However, as the temperature increases, the strength of the (standard size stainless steel tubing) nozzle decreases dramatically. Therefore, the required thickness of the nozzle walls was far greater than the injector wall thickness. The dimensions of the nozzle are listed in Figure 5.

Unlike the nozzle and injector dimensions, the length of the quartz tube is determined by the flame dimensions. Since the objective of this combustor is to observe burning aluminum particles, it is necessary that the length of the tube be sufficient for complete combustion of the particles to occur. Therefore, the maximum ignition and burn times were calculated for 25 μm particles. Using these times and the maximum fuel injection velocity, it was possible to determine the length required for the particles to ignite and burn. An additional length of quartz tubing was required for the inviscid core which occurs at the injector face (see Figure 6). With an added margin of error, the total quartz tube was determined to be 10 in. long. Since the inside diameter of the quartz tube must be the same as the inside diameter of the nitrogen injector tube (to eliminate recirculation zones near the quartz tube wall), an inside diameter of 0.93 in. was selected for the quartz tube. However, due to the high heat transfer rates and the poor strength of quartz in tension, it was necessary to design a two wall construction for the quartz combustion chamber (see Figure 7). The reasons for this two wall construction are discussed in the conclusions and recommendations section.
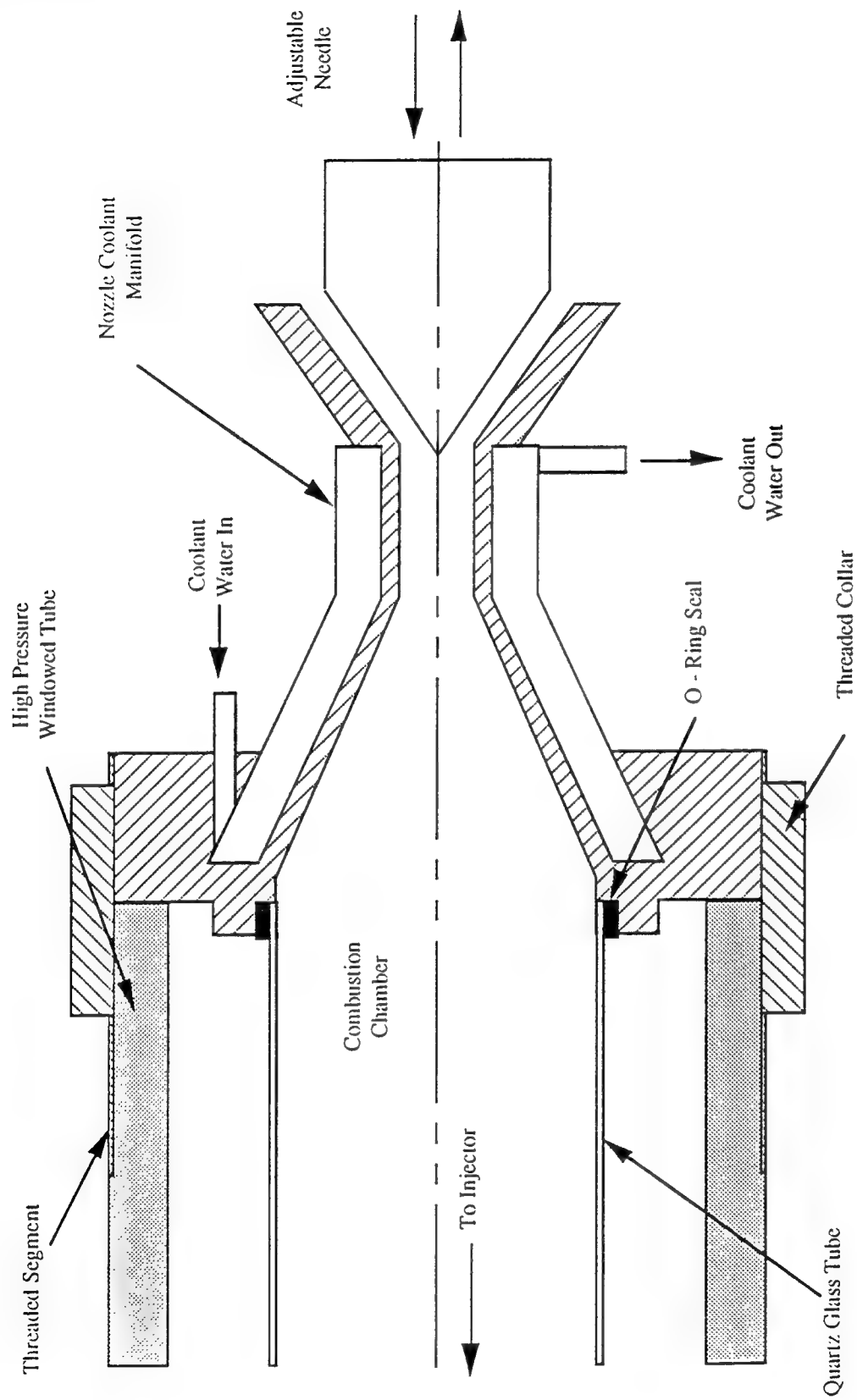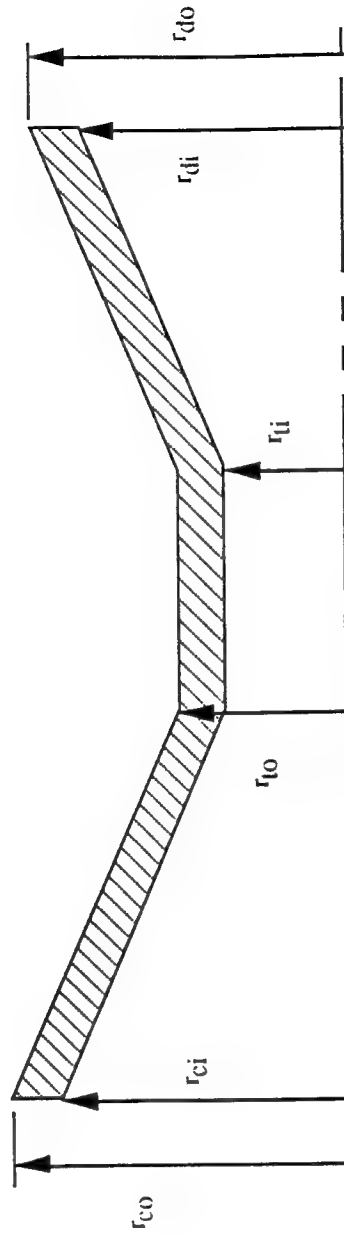
Figure 4: Nozzle Design Configuration

Note: Dimensions are in inches

r_{co}	r_{ci}	r_{to}	r_{ti}	r_{do}	r_{di}
0.5625	0.4675	0.078125	0.062	0.444	0.349

Figure 5: Nozzle Dimensions
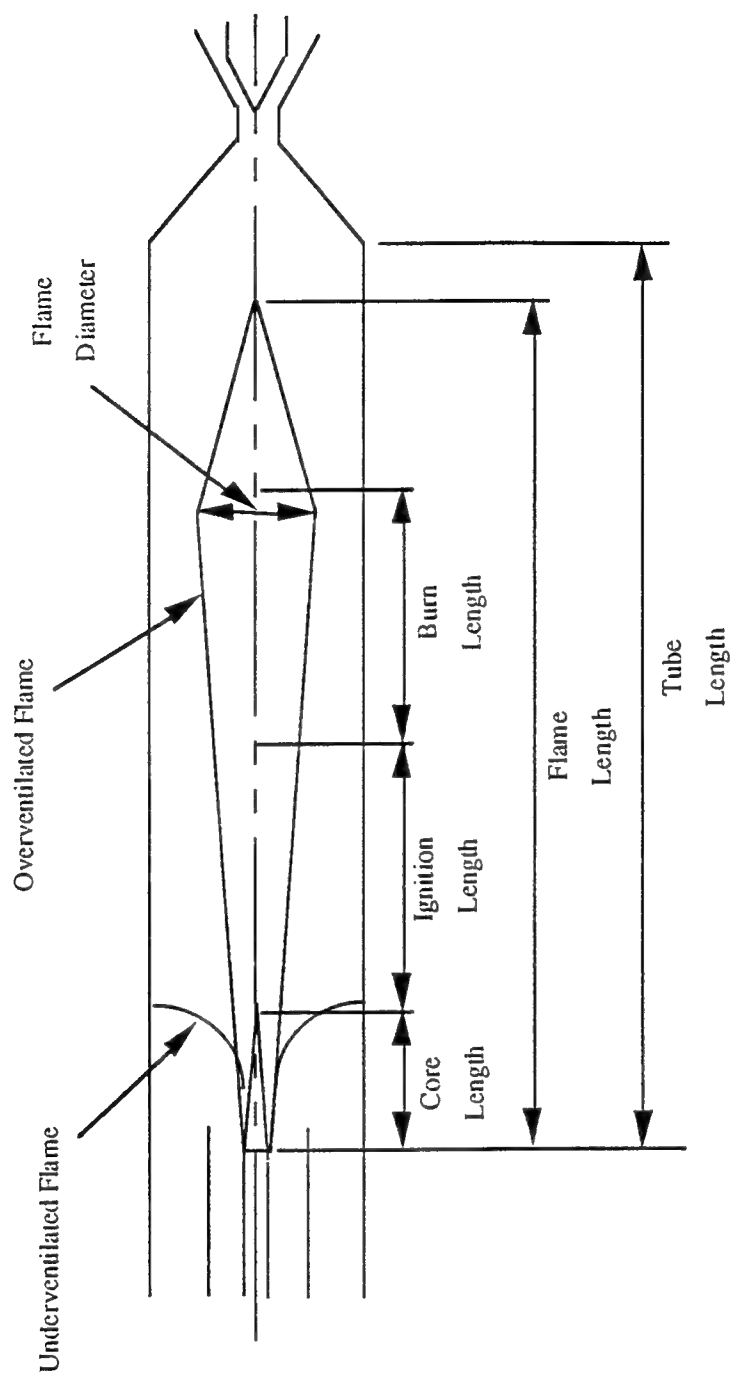
Figure 6: Simplified Flame Diagram
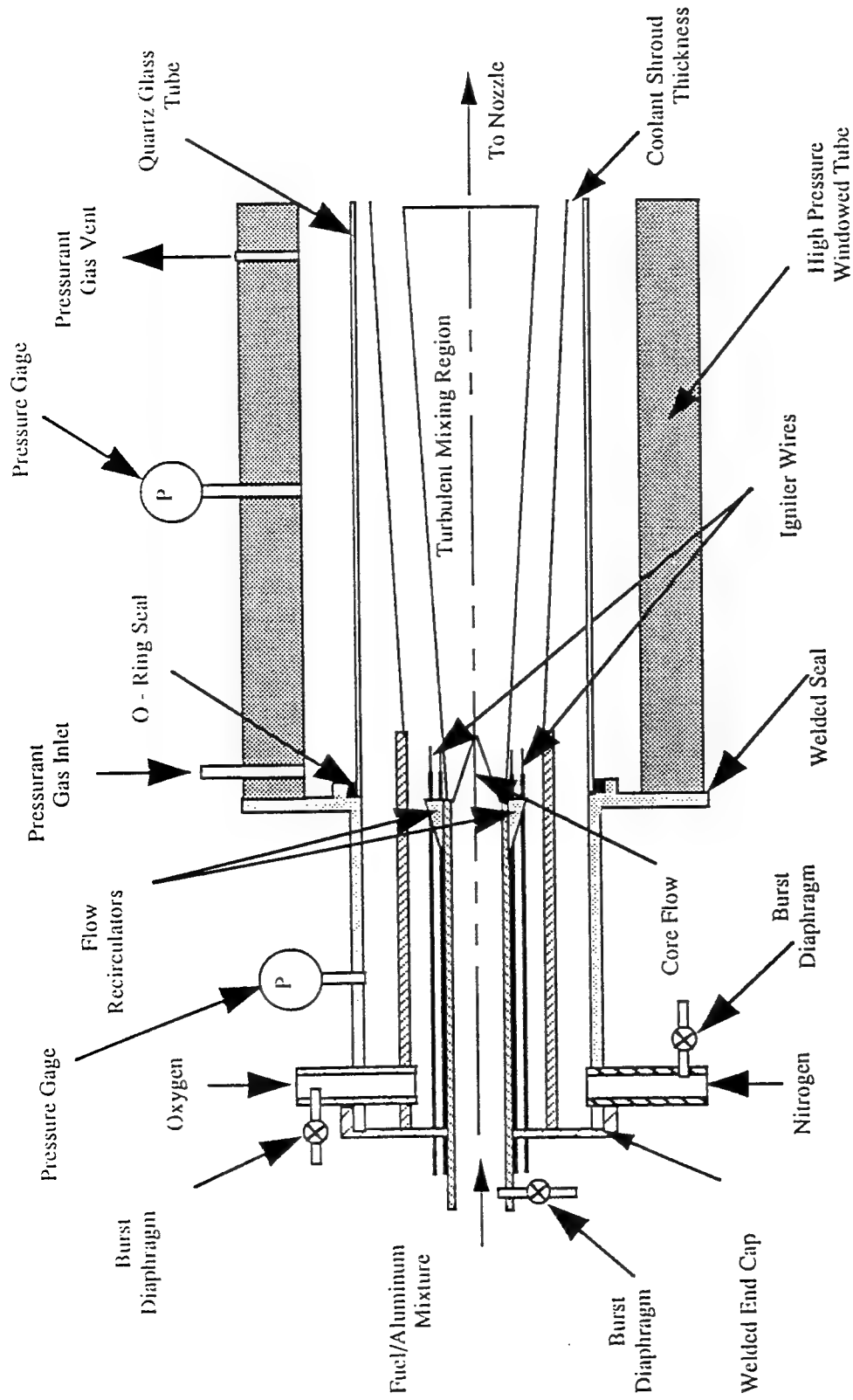
Figure 7: Injector Manifold Design

Quartz Glass Tube

To Nozzle

Coolant Shroud Thickness

Pressurant Gas Vent

High Pressure Windowed Tube

Pressure Gage

Turbulent Mixing Region

O - Ring Seal

Igniter Wires

Pressurant Gas Inlet

Flow Recirculators

Welded Seal

Pressure Gage

Core Flow

Oxygen

Burst Diaphragm

Nitrogen

Burst Diaphragm

Fuel/Aluminum Mixture

Burst Diaphragm

Welded End Cap

Conclusions and Recommendations

As stated earlier, the design of this combustor is intended to start as simple as possible and gradually work towards more complex designs that would more accurately simulate the combustion of aluminum in a solid rocket motor environment. This design work was mainly based upon simplified pressure vessel equations, equilibrium chemistry calculations, and a simplified heat transfer analysis. (Each of these design criterion are discussed below.)

Although the simplified pressure vessel equations are by no means a finite element analysis, they should be sufficiently accurate for the intended operations of this combustor. However, the dimensions of the central injector were arbitrarily chosen. In reality, the area and flow rate of each injection tube could drastically change the stability of the flame and the combustion products which are formed. In addition, this design assumed that the nitrogen and oxygen would not intermix. In reality, this is not the case. This is due to the fact that there will be some vorticity produced between the oxygen and nitrogen interface. Any substantial mixing between these two gases could cause the flame to expand to the quartz tube wall resulting in excessive heat transfer rates. For these reasons, it is imperative that the flow rates and flame stability be verified and all necessary redesign work be completed before the initial test operations begin. These redesigns may include changing the dimensions of the injectors, the flow re-circulator locations, and the flow rates of the gases.

The use of equilibrium chemistry for this combustor should provide a fairly accurate overview of the species, transport properties, and temperatures which will be produced in this combustor. The accuracy of the NASA equilibrium chemistry code has been confirmed in numerous tests. However, since these chemistry calculations assume infinite reaction rates and reactions which continue until completion, the actual species produced in this combustor will probably be slightly different from those which were predicted. In reality, the rate of reaction will be mainly controlled by the turbulent mixing of the species at

the injector face. Also, the reaction of the aluminum particles (with oxygen) may be difficult due to the (possible) lack of oxygen along the center of the combustor. Further complications which may affect the aluminum particle combustion include the formation of a protective oxide on the particles and the collision of molten aluminum particles which may form large agglomerates (and therefore, cause substantially longer burn times). All of these factors must also be experimentally verified and any necessary design changes must be incorporated into the combustor before actual tests can begin.

Another problem which may occur during the operation of this combustor is the erosion of the nozzle throat. Although the heat transfer calculations should insure that the nozzle will be maintained below its melting temperature, it does not necessarily guarantee that the nozzle throat will not erode. The high flow velocities of the reactive and corrosive combustion gases can still erode (and possibly react) with the stainless steel nozzle. Since the velocities should be their maximum at the nozzle throat, it is recommended that the nozzle design be incorporated with graphite inserts in the nozzle throat. Although the graphite insert will still probably erode, it should not threaten the integrity of the nozzle itself. This should, in the long run, decrease the cost and increase the safety of the nozzle design. Additionally, since the ceramic needle may cause the nozzle to plug with aluminum slag (resulting in catastrophic over pressures), it is recommended that the variable throat area design be discarded in favor of several different nozzles. Each of these nozzles would have graphite inserts with the throat area corresponding to choked flow for the flow rate of the particular experiment. Although this may be more expensive, it is necessary to insure the safe operation of this device.

Due to the high heat transfer rates it was necessary to use a two wall construction for the quartz observation windows (Figure 4 and Figure 6). This was also necessary due to the poor strength of quartz in tension (especially at high temperatures). If a single glass tube were used for this combustor, it would require a tube which would be approximately 0.5 in. thick for a 1.0 in. inside diameter tube. This thick tube would act as a (powerful) lens which would make data acquisition difficult. Alternatively, the combustor

could use a single chamber with a thick recessed quartz window. In order to cool this recessed window, it would be necessary to purge this window with a coolant gas. This may make it difficult to observe the combustion process along the entire length of the combustor. Also, this alternative design may result in the formation of recirculation zones of hot combustion gases near the combustor walls (unlike the two wall design) resulting in catastrophic heat transfer rates. Not only may this alternative design be more dangerous, but it could also be more expensive to develop the purge system for the recessed window design. For these reasons, it is recommended to develop the two wall design configuration.

Obviously, the combustion of aluminum particles in a flame may not accurately simulate the heterogeneous reactions which occur within an actual solid rocket motor. However, this combustor concept is designed to be capable of observing the effects of the individual species on the aluminum combustion process. This design should also be able to observe the effects of individual species on the optical properties of aluminum and its oxides. In order to accurately simulate the solid rocket motor species, it is necessary to operate this combustor in the fuel rich regime for the first two suggested fuels (hydrogen and methane). (The two main species produced in a typical rocket motor are free hydrogen and carbon monoxide.) Fuels which will be used in later experiments should produce hydrochloric acid and free nitrogen (two other species produced in solid rocket motors). It is recommended that each of these species and their effects on aluminum be studied individually (if at all possible). Eventually, after numerous limited specie tests, this combustor will be able to observe aluminum combustion in an environment which contains all of the prevalent species found in a solid rocket motor.

Once this combustor is built (after all of the necessary redesigns have been incorporated), it should provide a versatile research tool. Not only should this combustor provide insight into the aluminum combustion process, but it could also be a valuable tool for the prediction of coaxial jet mixing and combustion. This combustor may even be able to suggest new fuels and additives for solid rocket motors. In conclusion, this combustor (or its derivatives) should be a valuable research tool for years to come.

References

1. Geierman, R. J., "Design of an Aluminum Powder Combustor for Simulating the Combustion Conditions in Solid Rocket Motors", A Thesis Presented for the Master of Science Degree, August, 1993.

2. Gordon and McBride, "Computer Program for Calculation of Complex Equilibrium Compositions, Rocket Performance, Incident and Reflected Shocks and Chapman - Jouquet Detonations", NASA SP-273, February, 1973.

VALIDATION OF A RECENTLY DEVELOPED SOLID PARTICULATE DISPERSION MODEL

William Kvasnak

Department of Mechanical and Aeronautical Engineering

Clarkson University, Potsdam, NY 13699

VALIDATION OF A RECENTLY DEVELOPED SOLID PARTICULATE DISPERSION MODEL

William Kvasnak

Department of Mechanical and Aeronautical Engineering

Clarkson University. Potsdam. NY 13699

Abstract

Validation of a Lagrangian transport model for predicting solid particulate concentrations is presented. Confidence in the model is supported by a comparison of numerical computations from the LTM3D code developed at AEDC with analytical and experimental data. Exact solutions of the equations of motion are compared with numerical results in order to validate Brownian motion and gravitational sedimentation. A comparison with experimental data for particle deposition in a channel is used to validate the turbulent dispersion model. The instantaneous fluctuations are simulated as continuous Gaussian random vector fields. The Lagrangian equations of motion provide particle velocities induced by Stokes drag. Brownian diffusion, and gravity. An assumption of one way fluid particle interaction and no particle - particle interaction places a maximum on the particle concentration of less than 0.02% by volume. Three validation test cases are presented for comparison with analytical and experimental results. The exact solutions and data are in excellent agreement with the numerical results.

Introduction

Ecological problems induced by pollution have recently become a priority to the Air Force, and mankind in general. Difficulties caused by acid rain, smog, oil spills, and forest fires, have caused the Environmental Protection Agency (EPA) to place strict regulations on the total amount of emissions allowed within containment areas. AEDC has always operated within EPA guidelines, but as EPA regulations become more and more strict the need for atmospheric plume dispersion predictions become imperative to the AEDC test facility permit process. Under the 1992 Graduate Student Research Program the LTM3D solid particulate dispersion code was developed to help fulfill this need. The motivation and methodology used in the LTM3D model are detailed in Kvasnak [1]. This paper details the next step in the development, i.e., the validation of the code for simple test problems.

Because of the importance of solid particle dispersion in numerous industrial and environmental processes, much research has been done in the area of aerosol particle dynamics. Consequently, a wealth of knowledge exists on the subject. Fuchs [2], Hidy and Brock [3], van de Hulst [4], Twomey [5], and Cadle [6] are just a few of the classic textbooks on the subject. Particle dispersion in turbulent flows was simulated by Ahmadi [7], Li and Ahmadi [8], and Ounis et al. [9]. The overall goal of this project is to provide a large-scale simulation of atmospheric dispersion of fine particles. The use of a supercomputer facilitates large-scale simulations which may provide the needed insight into the processes that dominate the motion and dispersion of fine particles in the atmosphere. The report of Kvasnak [1] detailed the development of an efficient algorithm for this purpose; however, the Crank - Nicholson discretization has been replaced by a fully implicit backward-Euler scheme. This report details the validation and improvement of the LTM3D code in order to provide a useful validated and calibrated product for AEDC.

Review of the LTM3D Methodology

Kvasnak [1] describes in detail the methodology behind the LTM3D code. Here only a brief review is presented. In the simulation a steady-state flow field is taken from an incompressible Navier-Stokes equation solver. The Lagrangian equation of motion for a small particle is then solved in time for a release of particles. The concentration at every location in the flow is taken as a mass percent of particles in a given region, also called a cell. The distribution of particles is calculated on a local and global level. This method provides a time-accurate description of the dispersion of an emissive plume.

Mechanics of Aerosols

By definition, an aerosol is a suspension of solid or liquid particles in a gas. Common aerosols found in nature include: dust, smoke, fog, haze, and smog. Aerosols typically range in size from 0.001 to 100 microns in diameter. To place these dimensions in perspective, note that the mean free path of air is about

0.07 microns and that the wavelength of visible light is between 0.4 to 0.7 microns. Also, a diameter of 0.01 microns roughly corresponds to the transition limit between molecule and particle. Particles larger than 100 microns typically do not remain suspended in air for a significant time duration. Thus, the behavior of particles is significantly affected by their size.

Aerosol Particle Motion

The Lagrangian equation of motion for a Stokesian aerosol particle can be written as

$$m\frac{du^p}{dt} = \frac{3\pi\mu d}{C_c}(u^f - u^p) + mg,$$ (1)

where m is the mass of the particle and C_c is the Cunningham correction factor given by

$$C_c = 1 + \frac{2\lambda}{d}\left[1.257 + 0.4\exp\left(-\frac{1.1d}{2\lambda}\right)\right].$$ (2)

Dividing Eq. (1) by $3\pi\mu d/C_c$ yields

$$\tau\frac{du^p}{dt} + u^p = u^f + \tau g$$ (3)

where τ is the particle relaxation time. This is defined by

$$\tau = \frac{mC_c}{3\pi\mu d} = \frac{d^2\rho^p C_c}{18\mu}$$ (4)

where

$$m = \frac{\pi d^3}{6}\rho^p.$$ (5)

For relatively large particles, $C_c \approx 1$ and

$$\tau = \frac{d^2\rho^p}{18\mu}.$$ (6)

Simulation of a Turbulent Flow Field

The turbulent velocity field is constructed by the superposition of a fluctuating velocity onto a steady velocity field. In Kvasnak [1] a zero correlation Gaussian random variable with a strength $\vec{u'}$ was generated. The instantaneous velocity at the position of the particle was given as

$$\vec{u} = \vec{\bar{U}} + \vec{u'}$$ (7)

where $\vec{\bar{U}}$ is obtained from the Navier-Stokes solver. Then, a Gaussian random variable with strength equal to the rms velocity was used as a fluctuating component.

This procedure essentially models turbulence as a white-noise process. Kraichnan [10] suggested a simple method of generating a Gaussian random field which resembles isotropic turbulence. Accordingly, the instantaneous fluctuating velocity is given as

$$\vec{u'^*}(\vec{x^*},t) = \sqrt{\frac{2}{N}}\left(\sum_{n=1}^{N}\vec{u}_1(\vec{k_n})\cos(\vec{k_n}\cdot\vec{x^*} + \omega_n t^*) + \sum_{n=1}^{N}\vec{u}_2(\vec{k_n})\sin(\vec{k_n}\cdot\vec{x^*} + \omega_n t^*)\right).$$ (8)

In this equation, the quantities $\vec{u}_1$ and $\vec{u}_2$ are defined as

$$\vec{u}_1(\vec{k_n}) = \xi_n \times \vec{k_n} \tag{9}$$

$$\vec{u}_2(\vec{k_n}) = \zeta_n \times \vec{k_n}. \tag{10}$$

with

$$\vec{k_n} \cdot \vec{u_1}(\vec{k_n}) = \vec{k_n} \cdot \vec{u_2}(\vec{k_n}) = 0, \tag{11}$$

to ensure incompressibility. The quantities ξ, ζ, and ω are Gaussian distributed random variables with a standard deviation of 1. Each component of $\vec{k}$ is a Gaussian random variable with a standard deviation of $1/2$. The quantity N is the number of terms in the series. In this analysis, N was arbitrarily chosen as 100.

The nondimensional quantities are defined by

$$\vec{x^*} = \frac{\vec{x}}{l_o} \tag{12}$$

$$t^* = \frac{t}{t_o} \tag{13}$$

$$\vec{u'^*} = \frac{\vec{u'}}{u^*} \tag{14}$$

where l_o, and t_o are the integral length and time scales of turbulence, u^* is the friction velocity, and u' is the rms fluctuating fluid velocity which Kraichnan [10], assumed to be isotropic. For this pseudo-turbulent velocity field the energy spectrum $E(k)$ is given as

$$E(k) = 16\sqrt{\frac{2}{\pi}}k^4 e^{-2k^2}. \tag{15}$$

The analysis of Li and Ahmadi [8] provides an improvement based on the well-known fact of anisotropy in a turbulent shear layer. In order to estimate the anisotropy, they suggested the modification

$$\vec{u_i'} = \vec{u_i'}e_i(\vec{y}), \tag{16}$$

where $e_i(y)$ are empirically generated shape functions chosen to match the data of Kreplin and Ecklemann [11], reproduced in Figure 1.

Brownian Motion

Small particles suspended in a fluid undergo random translational motion due to molecular collisions. This phenomenon is referred to as Brownian motion. The Brownian motion leads to the diffusion of particles in accordance with Fick's law

$$J_i = -D\frac{dc}{dx_i}. \tag{17}$$

In this formula, c is the particle concentration, J is the particle flux, and D is the diffusion coefficient. The diffusivity is given as

$$D = \frac{kT}{3\pi\mu d} C_c.$$ (18)

The diffusion equation can be obtained by directly substituting Fick's law into the equation for mass conservation. The Brownian forces acting on the particle may be modeled as a white-noise process. The Brownian motion of particles is modeled by

$$\frac{du}{dt} + \beta u = n(t)$$ (19)

with

$$\beta = \frac{3\pi\mu d}{C_c m} = \frac{1}{\tau}.$$ (20)

Also, $n(t)$ is a Gaussian white-noise process with a spectral intensity

$$S_{nn} = \frac{2kT\beta}{\pi m}.$$ (21)

where k is the Boltzmann constant and T is the temperature. From Eq. (21) it follows that the power spectrum of the particle velocity is given as

$$S_{uu}(\omega) = \mid H(\omega) \mid^2 S_{nn}(\omega).$$ (22)

Here the system function, $H(\omega)$, is defined by

$$H(\omega) = \frac{1}{i\omega + \beta}.$$ (23)

Thus,

$$S_{uu}(\omega) = \frac{2kT\beta}{\pi m(\omega^2 + \beta^2)}.$$ (24)

The particle autocorrelation function is defined as

$$R_{uu}(\tau) = \overline{u'(t)u'(t+\tau)},$$ (25)

and may be found by taking the inverse Fourier transform of the power spectrum, i.e.,

$$R_{uu}(\tau) = \frac{kT}{m} e^{-\beta|\tau|}.$$ (26)

The diffusivity is then given by

$$D = \int_0^\infty R_{uu}(\tau)d\tau = \frac{kTC_c}{3\pi\mu d}.$$ (27)

Additional details of the Brownian motion analysis and the other aspects of particle motion and transport may be found in Ahmadi [7].

Results

In the following section three test cases are outlined. Each case was chosen to validate one of the individual mechanisms involved with atmospheric dispersion of solid particles.

Test Case 1: Terminal Velocity

The equation of motion of a small particle in a gravitational field is given by equation (3) and is reiterated here for convenience

$$\tau \frac{du^p}{dt} + u^p = u^f + \tau g. \tag{28}$$

If the vector velocity of the fluid is identically zero, (i.e., stagnant), then Eq. (28) reduces to

$$\frac{du^p}{dt} + \frac{1}{\tau} u^p = g. \tag{29}$$

Equation (29) is a linear first-order ordinary differential equation with constant coefficients and a constant forcing function. An equation of this type has a simple solution, which in this case is given by,

$$u^p = \tau g \left(exp^{-\frac{t}{\tau}} - 1 \right) \tag{30}$$

where u^p is the particle velocity, τ is the particle relaxation time, g is the acceleration due to gravity, and t is time. As the solution approaches steady state the velocity of the particle "relaxes" to a steady velocity equal to τg. This is commonly called the terminal velocity. Figure 2 shows a comparison of the computed numerical and analytical nondimensional velocity as a function of time. In this graph the velocity of the particle has been normalized by the terminal velocity. The curves show the excellent agreement between the theoretical and the computed numerical results. Three curves are shown to illustrate the effect of size on the terminal velocity, and the relaxation time of the particle. Here it is clearly seen that a $1\mu m$ particle relaxes to terminal velocity in less than 2×10^{-5} seconds, at which time it reaches a maximum velocity of about $2 \times 10^{-4} m/s$. A $10\mu m$ particle takes slightly longer, approximately 4×10^{-3} seconds, and has a maximum falling rate of $0.04 m/s$. A $100\mu m$ particle will not reach terminal velocity until 0.07 seconds and will fall at a rate of $0.69 m/s$. From these results, it is easy to see why a $1\mu m$ or a $10\mu m$ particle will stay suspended in the atmosphere for long periods of time, whereas a $100\mu m$ particle will most likely precipitate out quickly under the force of gravity.

Test Case 2: Brownian Diffusion in a Rectangular Box

In this test case the Brownian dispersion model is validated. The discrete particles are replaced by a continuous function representing the dispersed matter. The Eulerian equation of conservation of species is given as

$$\frac{\partial c}{\partial t} = D \left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} + \frac{\partial^2 c}{\partial z^2} \right) \tag{31}$$

It is postulated that the solution of Eq. (31) and the Lagrangian equations of motion should give equivalent results for a large number of infinitely small particles.

For a point source in a rectangular box, the time history of the dispersion process is given by the conservation of species equation with delta function forcing terms. Specifically,

$$\frac{\partial c}{\partial t} - D \left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} + \frac{\partial^2 c}{\partial z^2} \right) = q\delta(x - x_o)\delta(y - y_o)\delta(z - z_o)\delta(t - t_o) \qquad (32)$$

For all homogeneous boundary conditions this equation has an exact Fourier series solution given by

$$c = \sum_n \sum_m \sum_p \frac{8q}{l_x l_y l_z} \sin(\frac{n\pi x_o}{l_x}) \sin(\frac{m\pi y_o}{l_y}) \sin(\frac{p\pi z_o}{l_z})$$

$$exp^{-D \left(\frac{n^2\pi^2}{l_x^2} + \frac{m^2\pi^2}{l_y^2} + \frac{p^2\pi^2}{l_z^2} \right)(t - t_o)} \sin(\frac{n\pi x}{l_x}) \sin(\frac{m\pi y}{l_y}) \sin(\frac{p\pi z}{l_z}) \qquad (33)$$

where l_x, l_y, and l_z are the lengths of each side of the box in the x-, y-, z-direction respectively, and x_o, y_o, z_o are the x-, y-, and z-positions of the release, respectively.

In order to approximate this solution as closely as possible, the simulation is done using an extremely fine mesh. The mesh is rectangular and the spacing between nodes is $10\mu m$. It should be noted that although the mesh spacing is small, the particle diameter of $0.01\mu m$ is still 1,000 times smaller than the mesh spacing. The cloud resulting from an instantaneous release of 3,000 particles after 500×10^{-5} seconds is shown in Figure 3 along with the computational mesh. In this test case the x-, y-, and z-directions are chosen arbitrarily. Figures 4 through 6 detail the particle dispersion in time. Figure 4 shows the cloud dispersion in the x-direction. Excellent agreement is shown between the analytical and computational results at small times; however, as time increases the numerical scheme overpredicts the analytical solution slightly. Figure 5 shows the dispersed cloud concentration in the y-direction. Again, the computed numerical solution slightly overestimates the dispersion of the cloud. Figure 6 shows the trend of overprediction in the z-direction of the cloud dispersion. The overpredictions are on the order of 5% in the worst case.

The computed numerical results have been shown to match well with the analytical results. If an infinite number of massless particles were released, and if small disturbances could reach infinity instantly, then the two results would match exactly. Errors in this calculation arise from the method used to estimate the concentration, and the inaccuracy of the continuum assumption implicit in the Eulerian reference frame. Incidentally, the over-estimation of the dispersion gives a more conservative estimation of pollutant dosage.

Test Case 3: Deposition Rate in a Turbulent Channel

In the previous test cases, exact solutions to the equations that describe the gravitational sedimentation, and Brownian dispersion were given. No comparisons with experimental data have been presented. The

validation of the turbulent dispersion cannot be solved analytically. There is also a lack of experimental data detailing solid particle dispersion in a turbulent flow. However, there exists some experimental data relating the deposition rates of particles in turbulent channel flow. Therefore, a test case was chosen to validate the LTM3D code with experimental data and test the turbulent dispersion model used.

The flow field used in the test case is a three-dimensional representation of a two-dimensional channel. A $21 \times 29 \times 6$ rectangular mesh, in the streamwise, vertical, and spanwise directions, respectively, was used in this test case. The mean flow field in a turbulent channel was given by Ahmadi et al. [12] as

$$\frac{\bar{u}}{\bar{u}_o} = \frac{1 - \eta^2 - \frac{k'_v}{k'} \frac{\cosh(k'\eta)}{\sinh k'} + \frac{k'_v}{k'} \frac{\cosh k'}{\sinh k'}}{1 + \frac{k'_v}{k'} \frac{\cosh k' - 1}{\sinh k'}} \tag{34}$$

where $\eta = 1 - y/h$ is the nondimensional distance from the centerline, h is the channel half-height, k'_v, and k' are nondimensional parameters defined as

$$k'_t = \frac{0.00703 Re^{0.763}}{1 - 0.71 Re^{0.0134}} - 2 \tag{35}$$

$$k' = k'_v \frac{1 - 0.71 Re^{0.134}}{0.71 Re^{0.0134} - 0.5}. \tag{36}$$

Here, $\bar{u}_o$ is the mean centerline velocity and is given by

$$\bar{u}_o = \frac{V}{0.71 Re^{0.0134}} \tag{37}$$

where V is the mean channel velocity and

$$Re = \frac{2\rho V h}{\mu} \tag{38}$$

is the Reynolds number, which in this case was set equal to $17,700$. The quantities ρ and μ are the fluid density, and dynamic viscosity, respectively. Figure 7 shows the mean velocity profile for a $25.4mm$ channel with a mean velocity of $5m/s$. At this Reynolds number the friction velocity is $0.3m/s$. The anisotropy of the turbulent fluctuations of the channel flow that was used in this test case is shown in Figure 8. The normal component of the fluctuations near the wall have a profound effect on the deposition rate of particles. It is well known that v' has a quadratic variation at short distances from the wall, i.e.,

$$v' = Ay^2 \tag{39}$$

where A is a constant empirically chosen to be 0.012 for this study.

Figure 9 shows the nondimensional deposition velocity of particles as a function of nondimensional particle relaxation time. For a uniform concentration of C_o near the wall of the channel, the deposition velocity is defined as

$$u_d = J/C_o \tag{40}$$

5 - 9

where J is the particle flux to the wall per unit time. The nondimensional deposition velocity is given as

$$u_d^+ = u_d/u^*. \tag{41}$$

The nondimensional particle relaxation time is defined to be

$$\tau^+ = \frac{Sd^2 u^{*2} C_c}{18\nu^2} \tag{42}$$

In Figure 9, the present simulation results are compared with the accumulation of experimental data as produced in Papavergos and Hedley [13]. Although there is substantial scatter in the data, the present computed results agree within the scatter of the data. The predictions are on the high edge of the experimental data ensuring a conservative estimate of deposition rate.

Conclusions

The computed results predicted by the LTM3D solid particulate dispersion code has been compared to the results from three well-behaved test cases. The terminal velocity, Brownian motion, and turbulent dispersion models have been tested using two analytical and one experimental tests. The terminal velocity test case had excellent agreement between the analytical and computed numerical results. An error of less than 1% was shown between the two solutions. The Brownian diffusion test case had slightly more error, approximately 5%, but it too offered significant confidence in the method. The deposition velocity calculation was shown to be in good agreement with the experimental data, although the data has significant scatter. These results are very encouraging and provide a higher degree of confidence in the code's predictive capability.

Suggested Future Work

The next step in the development of the predictive capability should be a study of the spatial and temporal resolution that is required to resolve the second-order differences in the particle motion versus the passive scalar method. The inclusion of temperature and density variations should be integrated into the model. Another logical improvement for future capability is to integrate the LTM3D code and the Lawrence Livermore National Laboratories FEM3A/B code to solve for gaseous, liquid, and solid particulate dispersion simultaneously. The combustion and chemical reaction of solid particles also may need to be accounted for.

Acknowledgements

References

1. Kvasnak, W., 1992, AFOSR Final Report, AEDC, Arnold Air Force Base TN.

2. Fuchs, N. A., The Mechanics of Aerosols, 1964, Pergamon Press Ltd., Oxford, Eng.

3. Hidy, G.M., and Brock, J.R., The Dynamics of Aerocolloidal Systems, 1970, Pregamon Press, Ltd., Eng.

4. van de Hulst, H.C., Light Scattering by Small Particles, 1957, Wiley, New York, N.Y.

5. Twomey, S., Atmospheric Aerosols, 1977, Elsevior Scientific Publishing Co., New York, N.Y.

6. Cadle, H.D., Particles in the Atmosphere, 1966, Reinhold Publishing Co., New York, N.Y.

7. Ahmadi G., Second Int. Conf. on Pneumatic transport of Solids in Pipes, BHRA, Bedford, U.S.

8. Li, A., and Ahmadi, G., 1991, Aerosol Science and Technology.

9. Ounis, H., Ahmadi, G., and McLaughlin, J., 1991, J. of Colloid and Interface Science.

10. Kraichnan, R.H., 1970, Phys. Fluids, 22.

11. Kreplin, R.H., and Ecklemann, H., 1979, Phys. Fluids, 22.

12. Ahmadi, G., Goldschmidt, V.W., and Dean, B., 1976, Iranian j. of Sci. and Tech., 5.

13. Papavergos, P.G., and Hedley, A.B., 1984, Chem. Eng. Res. Des., 62.
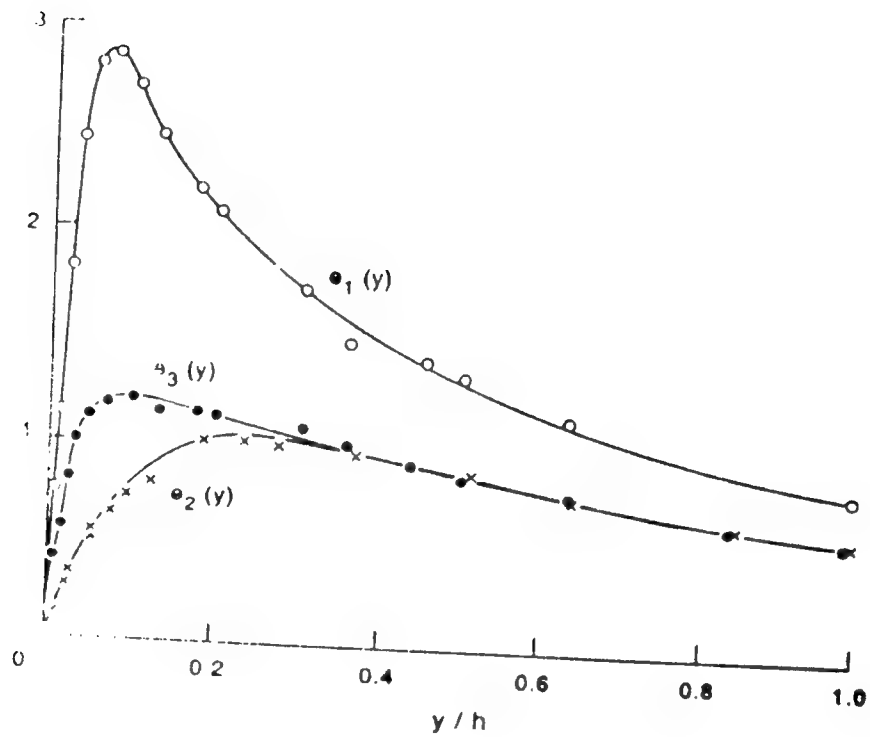
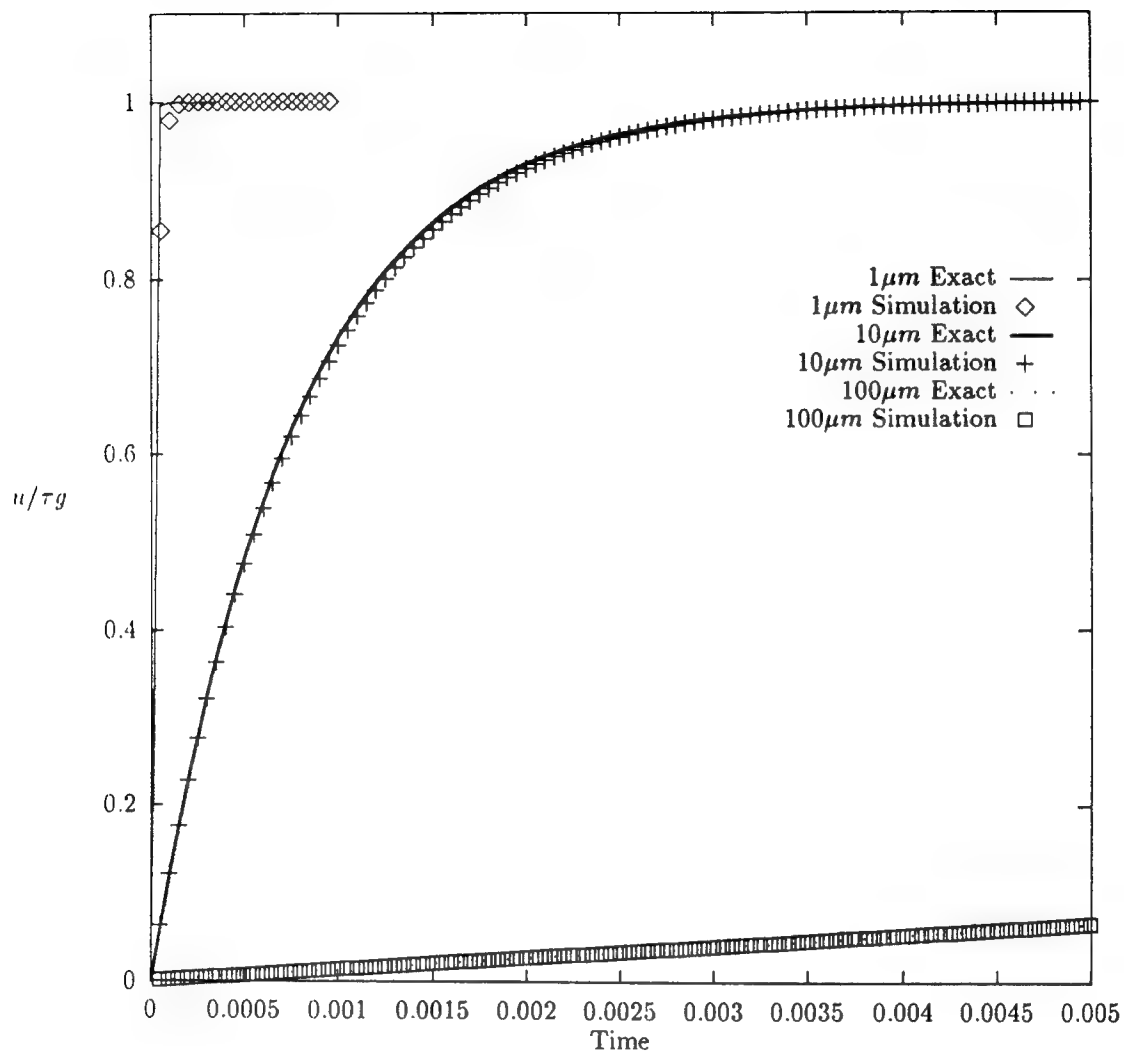Figure 1: Anisotropic Turbulence in a Channel, From Kreplin and Ecklemann [11]

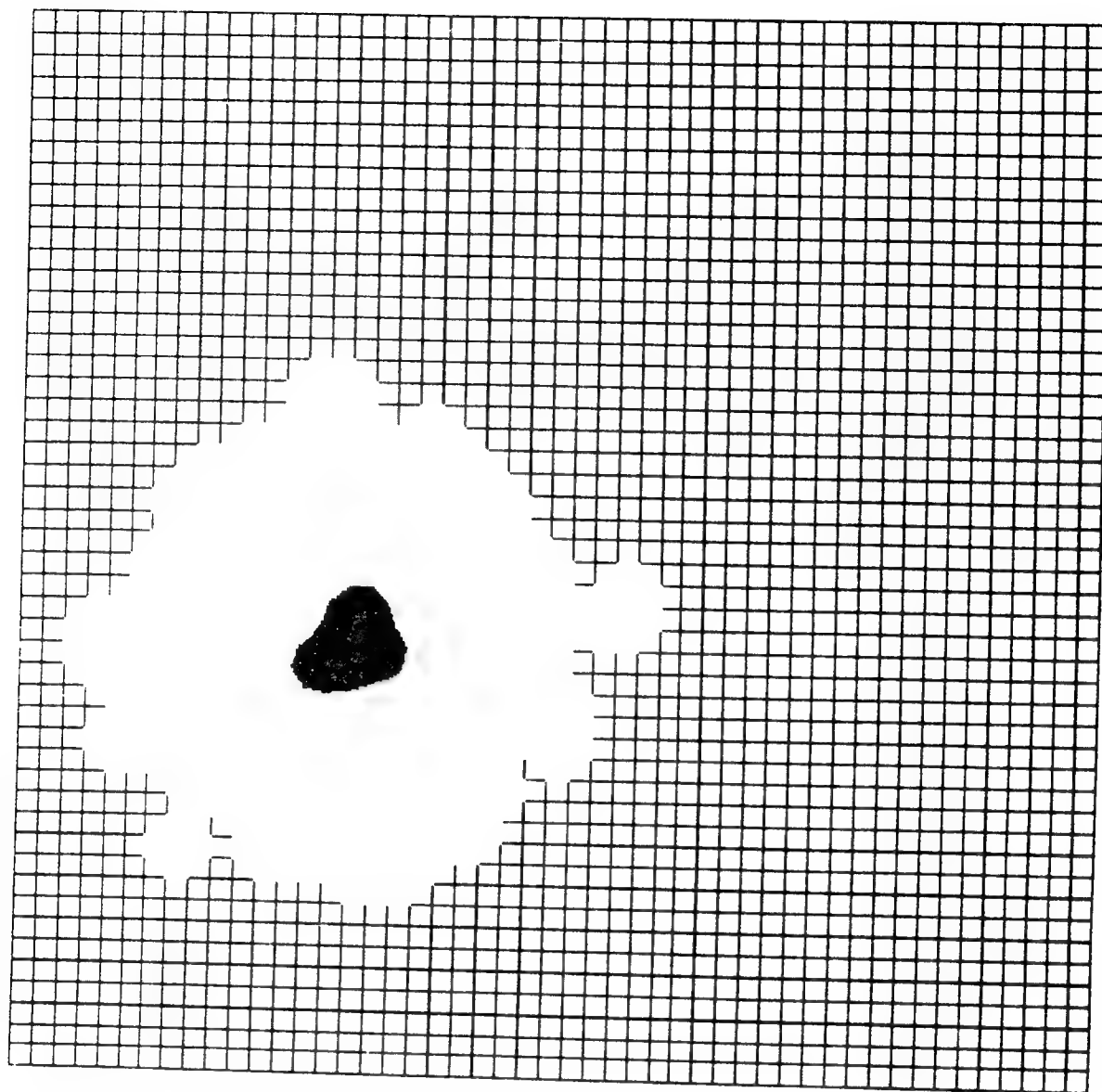Figure 2: Comparison of Computed Numerical and Analytical Terminal Velocities

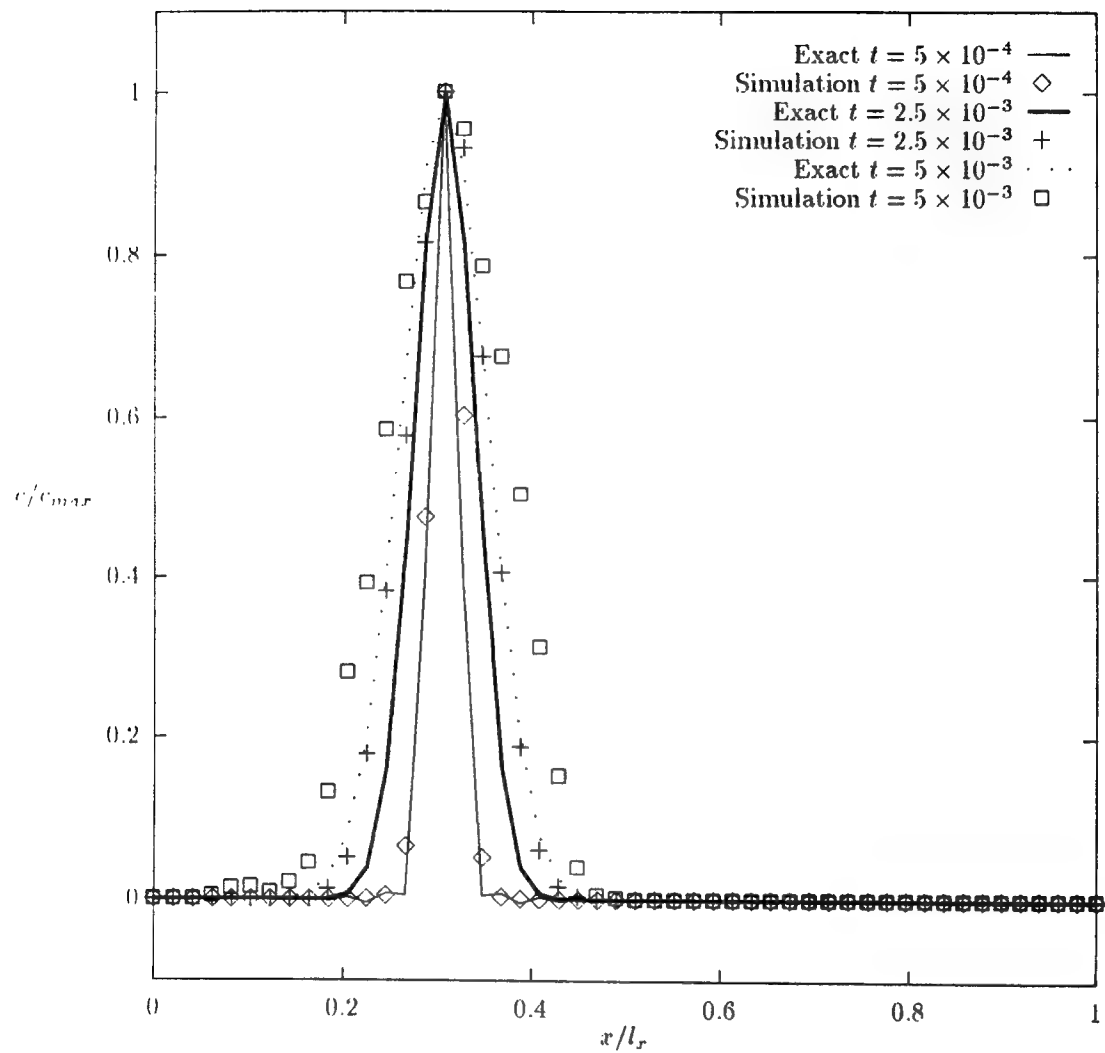Figure 3: Dispersion of 0.01 μm Particles in a Box

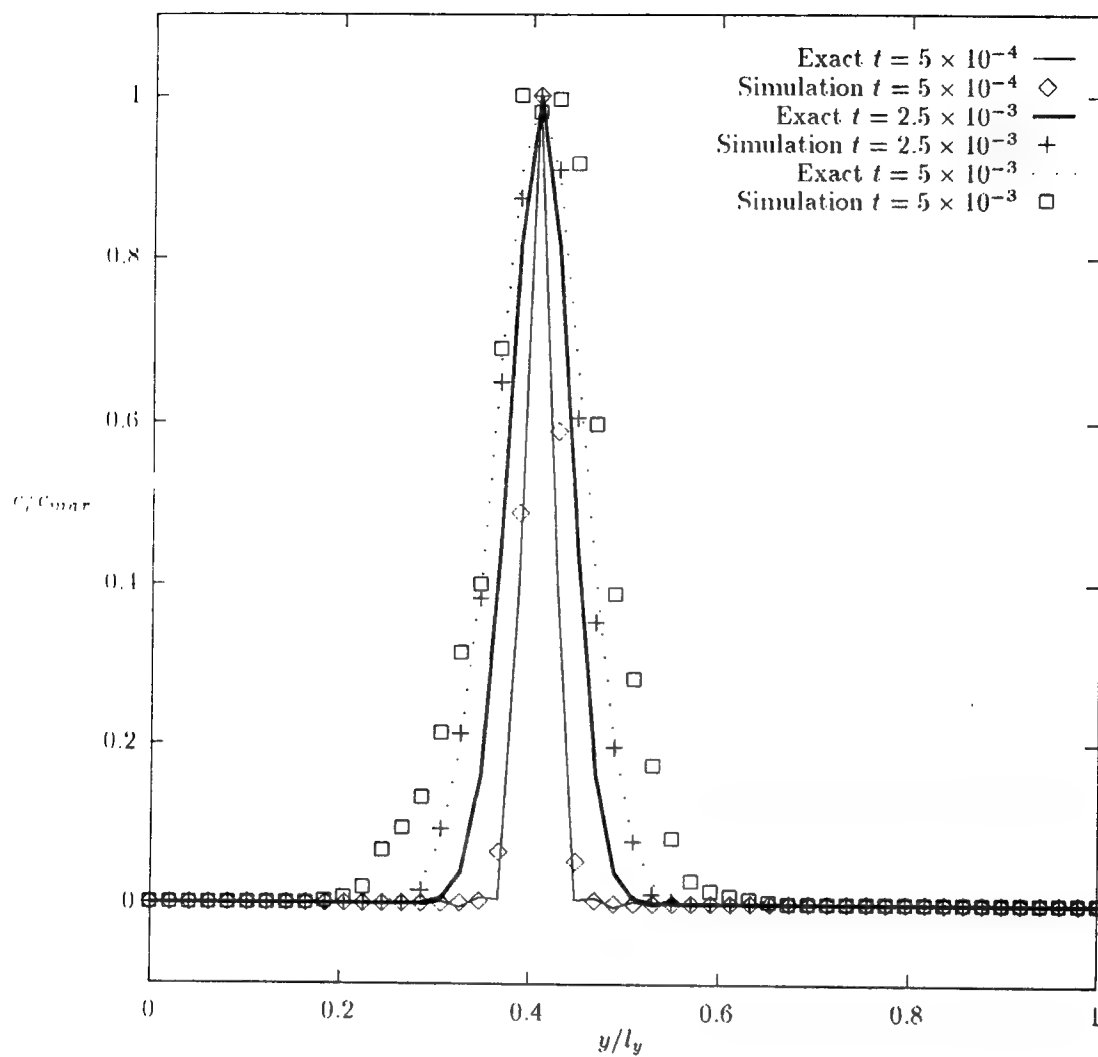Figure 4: Comparison of Particle Dispersion in x-Direction

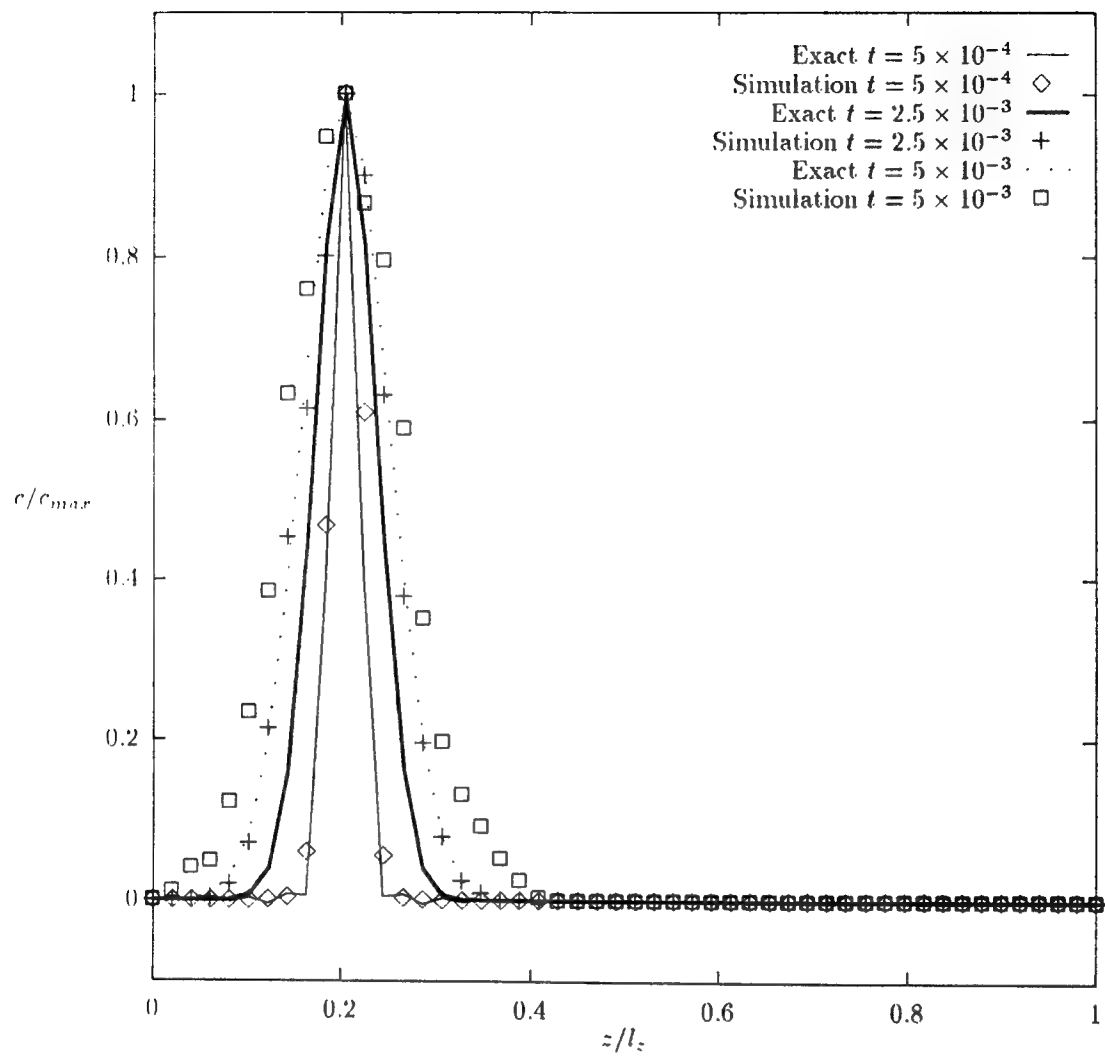Figure 5: Comparison of Particle Dispersion in y-Direction

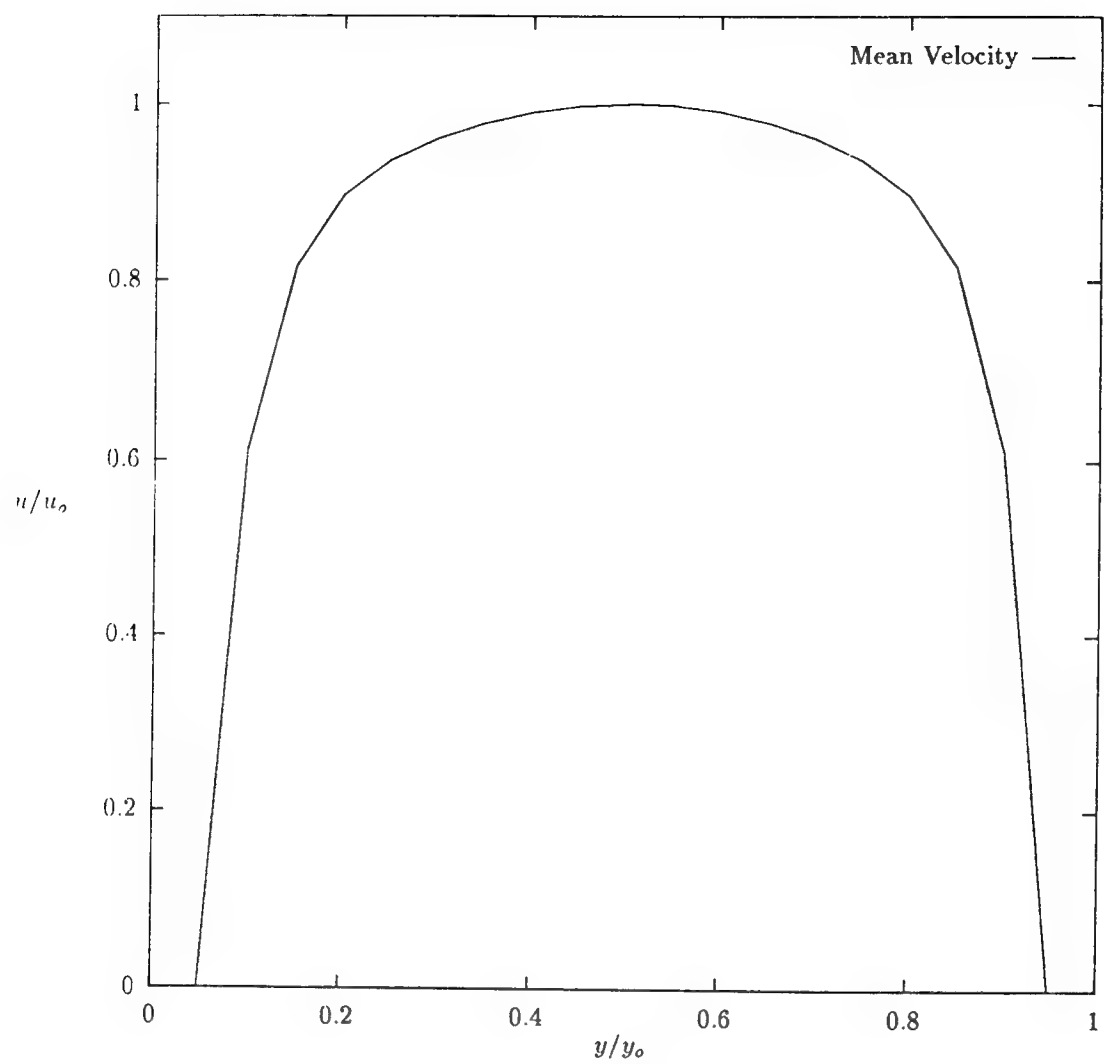Figure 6: Comparison of Particle Dispersion in z-Direction
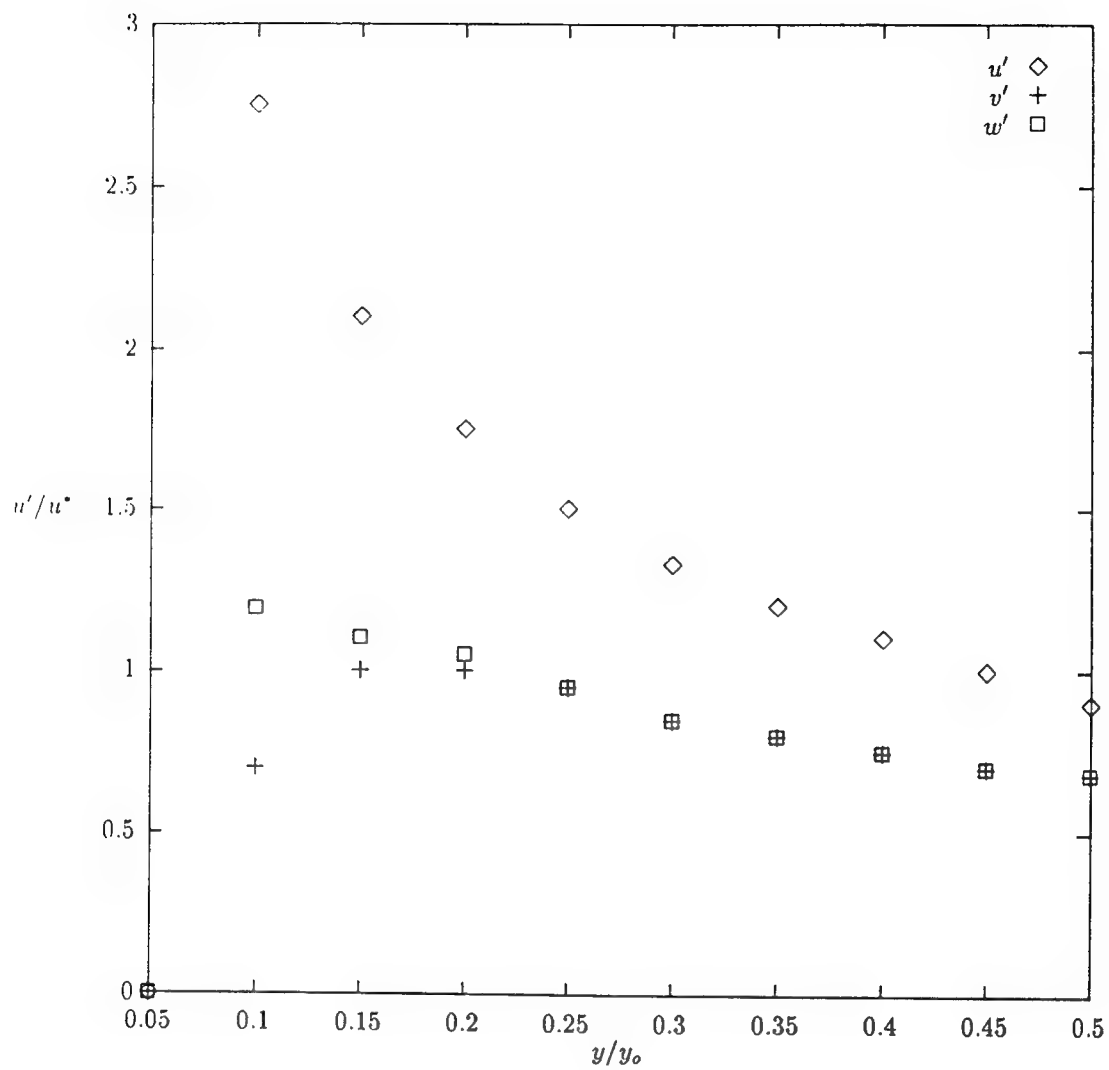
Figure 7: Mean Velocity Profile
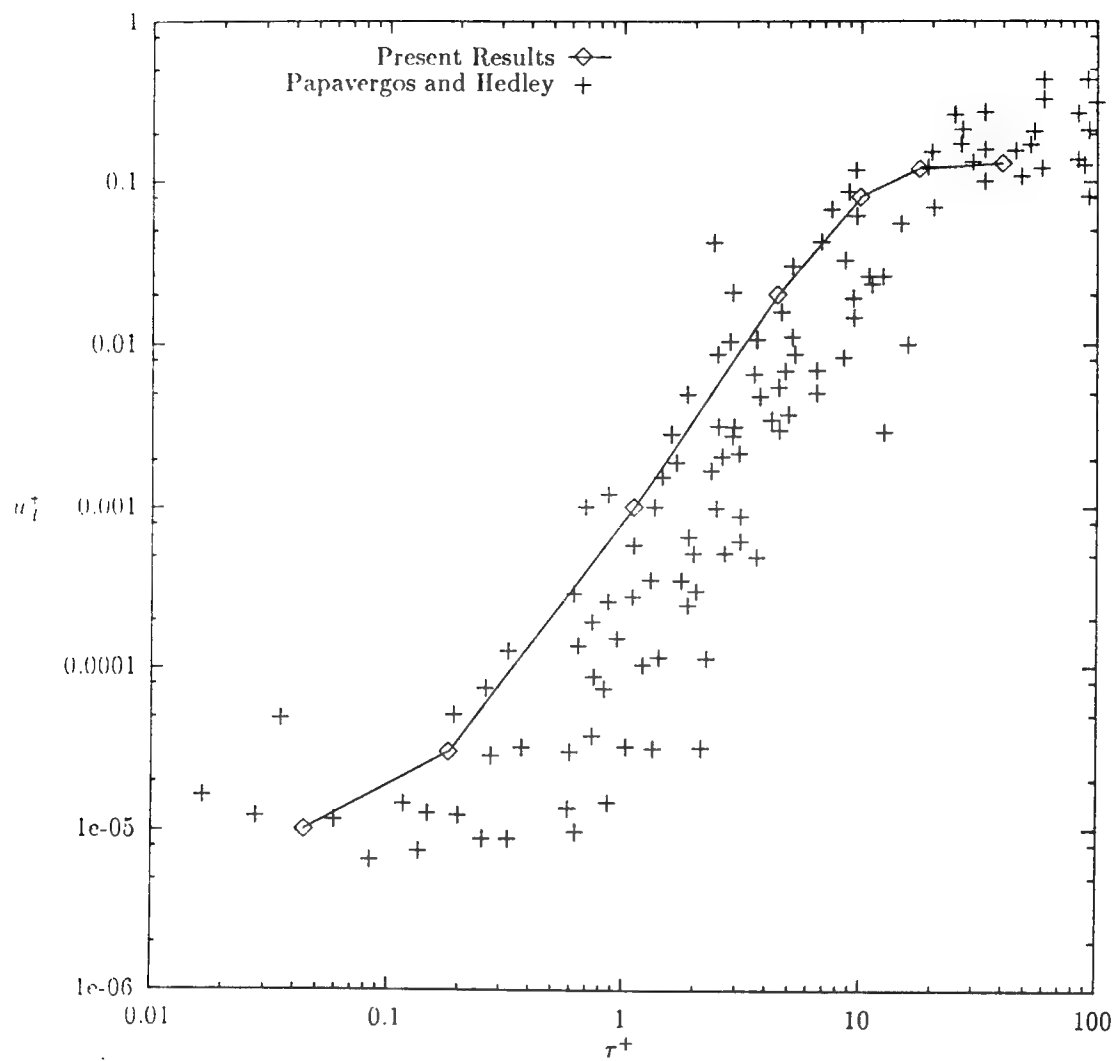
Figure 8: RMS Velocity Profile

Figure 9: Particle Deposition Velocities In a Turbulent Channel

A MODEL BASED PROGRAMMING ENVIRONMENT
FOR PARALLEL IMAGE PROCESSING

Michael S. Moore
PhD. Candidate
Department of Electrical Engineering

Vanderbilt University
400 24th Ave. S.
Nashville, TN 37235-1824

Final Report for:

Summer Research Program
Arnold Engineering Development Center

Sponsored by:

September 1993

A MODEL BASED PROGRAMMING ENVIRONMENT
FOR PARALLEL IMAGE PROCESSING

Michael S. Moore
PhD. Candidate
Department of Electrical Engineering
Vanderbilt University

Abstract

This report presents the results of the 1993 AFOSR–RDL Graduate Student Summer Research Program. The eventual goal of this continued research project is to create a high performance, model–based parallel image processing system with a graphical modeling environment. The main components needed to implement such a system are a modeling paradigm (language), a graphical model building environment, a model interpreter, a parallel run–time environment, and an image processing application library.

Parallel Processing Modeling Language (PPML) and the PPML interpreter provide a modeling environment that is adequate for modeling image processing algorithms. Khoros, a widely used image and signal processing system, has a graphical programming environment which can be used as a graphical model building environment. Khoros also provides a large, well developed library of image processing algorithms which can be re–used. In order to achieve the desired performance, we chose to use Multigraph as the parallel run–time environment. Multigraph is a parallel system integration tool developed at Vanderbilt University that allows the building of complex parallel applications from simpler processing blocks. By using the image processing libraries and graphical programming environment of Khoros in conjunction with the PPML modeling paradigm and the Multigraph parallel execution environment, we have created a model–based parallel image processing system which features a graphical programming environment and rich image processing library due to Khoros and concurrent execution due to Multigraph. The system has been tested for a large variety of algorithms and data–flows, and is considered a success.

Introduction:

At Arnold Engineering Development Center, there is a need for an image processing system with the high performance of parallelism as well as the flexibility and simplicity of visual programming. Usually, these requirements tend to produce contradicting results. While parallel implementations of algorithms exibit superior run-time performance, they are generally much more complex and require more programming time. It would be highly desirable to create an image processing programming environment which allows the user to create parallel algorithms while insulating him (or her) from the complexity involved in writing parallel programs. We have chosen to take a model-based approach to creating just such a programming environment. The various aspects of the algorithm are described by the user in the form of graphical pictures, or models. In the next stage, these models pass to an interpreter, which synthesizes the software necessary to implement the algorithm. The application is then sub-divided into parallel tasks, which are allocated to a hardware network and executed under the control of a parallel execution environment. The main components necessary for this system are (see figure1)

- A graphical model building environment

- An image processing modeling language and model interpreter

- A parallel execution environment

- A library of image processing functions

Khoros, a widely used image and signal processing system, has a graphical programming environment called Cantata in which the user creates image and signal processing applications by drawing pictures that model the signal flow of the algorithm. This graphical programming environment is easy to use and has become very popular. For this reason, we chose to use Cantata as the graphical modeling builder for the new system. Also, we chose to take advantage of the large Khoros image processing library, which contains a rich assortment of commonly used algorithms. Re-using the Khoros programs and libraries has saved huge amounts of development time. Since Khoros does not contain facilities to efficiently parallelize algorithms, its run-time performance is not adequate. Thus, we chose Multigraph as the parallel execution environment. Multigraph, a parallel system integration tool developed at Vanderbilt University, allows the building of complex algorithms from simpler processing blocks. It provides the underlying control and communication necessary
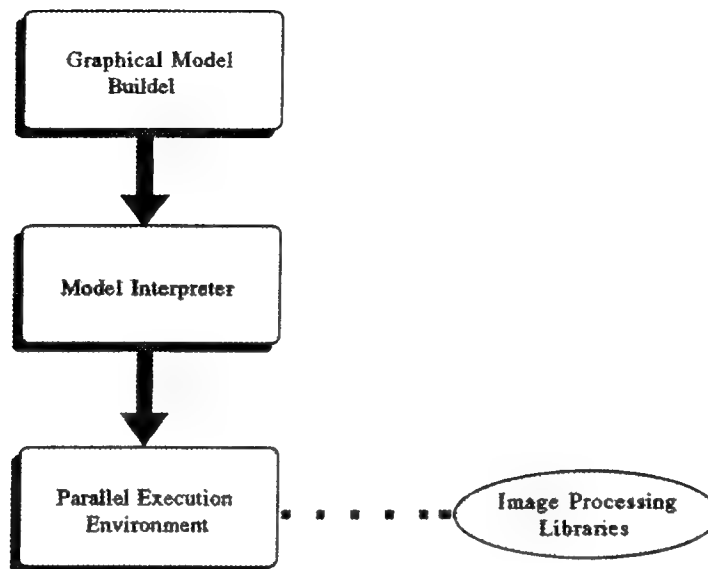
System Components

Figure 1: A Model–Based Parallel Image Processing System

for parallel execution, and is capable of efficiently distributing processes across a non–homogeneous network containing a variety of computer architectures. It has been shown that with the use of Multigraph, the natural parallelism inherent to image processing can be exploited, increasing dramatically the run–time performance of many algorithms [4] [5] [8] [6] [7]. Parallel Processing Modeling Language (PPML) and the Multigraph compatible PPML interpreter, which were developed at Vanderbilt University, were used as the image processing modeling environment for the system.

To achieve a hybrid Khoros–Multigraph system, a two level interface was built. Level 1 is the model level. A model translator interfaces the Cantata visual programming environment to the PPML model interpreter. Level 2 is the program level. A program converter acts as an interface between the Khoros image processing programs and the Multigraph execution environment.

The working system consists of the following components (refer to figure2) :

1. The Cantata Visual Editor

2. The Cantata to PPML model translator

3. The PPML model interpreter

4. The Khoros program to Multigraph actor converter

5. The Khoros image processing library

6. The MGK parallel execution environment
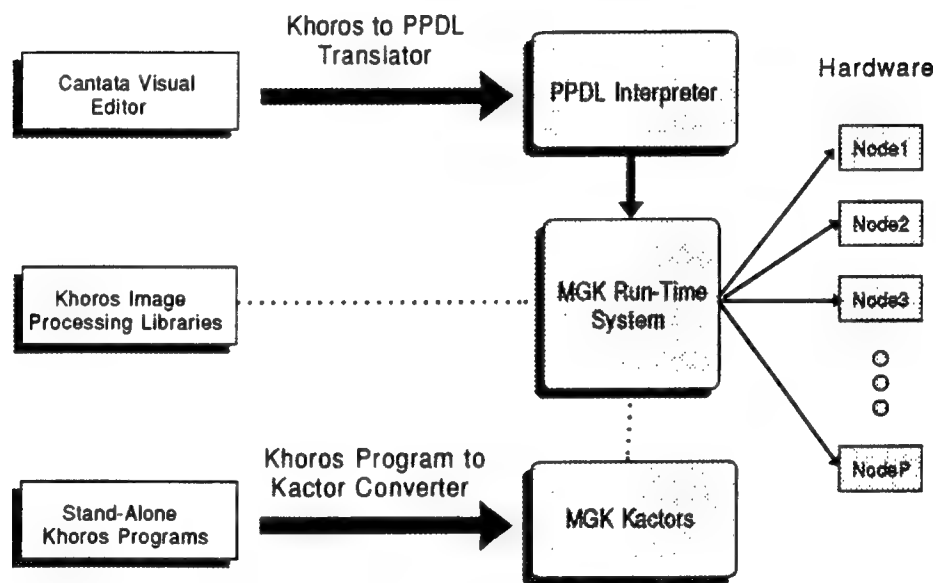
Khoros-Mgk System

Figure 2: Khoros–Multigraph Parallel Image Processing System

By interfacing Khoros with Multigraph, we have created an image processing system that has the run–time performance of Multigraph as well as the rich image processing library and graphical programming environment of Khoros.

In the following text, an explanation of the objectives of the summer research is given. Then follow descriptions of the image processing modeling paradigm and the model interpreter. Next, the details of the Khoros–Multigraph interface are described, including the model level interface and the program level interface. Lastly, the results are discussed and recommendations for future research are made.

Objectives Of The Research Effort:

The basic research objectives were:

- Create a program level interface between Khoros and the Multigraph execution environment.

- Create a model level interface between the Cantata visual programming environment and the PPML model interpreter.

- Integrate the system to create a model-based parallel image processing programming environment.

The Program Level Khoros–Multigraph Interface:

The reason for interfacing the Khoros image processing functions to the Multigraph kernel was to enable us to take advantage of the vast image processing libraries of Khoros without having to rewrite the programs. By re–using the Khoros programs, we have avoided wasting time and resources to "re–invent the wheel". Before describing the program converter, it is necessary to understand some of the details of both the khoros image processing environment and the Multigraph kernel.

The Khoros Image Processing Environment

The Khoros image processing environment includes a large array of commonly used image processing algorithms. Each of these algorithms is implemented by a stand–alone C program which can be executed from the command line or from within the Cantata visual programming environment. With either execution method, arguments are passed to the programs through a command line string. The Khoros programs read and write image files, the names of which appear after keywords in this command string. In Cantata, the image files provide the inter–program communication necessary to implement the data–flow. Each connection in the data–flow graph corresponds to an image file with a unique filename.

The structure of the Khoros programs is very regular. Each program has a customized argument parser which constructs a program specific argument data structure from the command string. This structure contains algorithmic parameters, flags, and input/output

filenames. The program checks the validity of the arguments and ensures that the input files exist. The input images are read from the files and checked for data type compatibility with the algorithm. The program then makes a call to the appropriate image processing library function with the algorithmic parameters and flags. On return, it writes the the output image(s) to a file(s). Thus, each program is merely an argument parsing and error checking mechanism that serves as an interface to an image processing library function.

The Multigraph Environment

The computational unit in the Multigraph environment is called an *actor*. Each actor has an I/O interface, a script, and a context (see [1]). The I/O interface consists of two kinds of *ports*, input and output ports, through which the actor can receive or propagate data. The port communication is implemented by an underlying message passing mechanism of the kernel. The script is the code that will be executed when the actor is activated, and is usually on the subroutine level. The execution of an actor script may have either an IFANY or an IFALL triggerring mode. An actor with IFALL triggering mode will be triggered only when data is present at all of its input ports, while an actor with IFANY triggering mode will be activated if data is present at any of one its input ports. The actor's context is a static memory area available to the script which is protected by the kernel. It can be used for passing parameters to the script or for storing state information.

The Kactor Program Interface

My approach was to create a protocall by which any khoros stand–alone program could be converted to a Multigraph actor. Pains were taken to make the converter work without any required editing of the khoros program sources, so as to minimize the depth of under-standing and effort required to install and maintain the system. The program conversion utility is called "kactor", for khoros–actor. Conversion is performed by recompiling the un–edited khoros source programs with the macros in "kactor.h" included and then linking to "libkactor.a" (in addition to the normal khoros support libraries). To include the macros without editing the source files, I copied the files "vinclude.h" and "xvinclude.h" to the kactor directory and added the following lines to them:

```
#ifdef MGK_VERSION
 #include "kactor.h"
#endif
```

Since either vinclude.h or xvinclude.h is included by every khoros program, this technique

does the trick. The kactor macros perform the following:

1. "main" is replaced by a standard IFALL actor script which first calls initialization fuction called "kactor–init". kactor–init parses the strings contained in the context structure into the argument vector used by the khoros program and creates tables used by mgk_get_image and mgk_send_image to map filenames to I/O port identifiers. The standard actor context structure includes three strings: (1) a command string (2) an input file keyword list (3) an output file keyword list. The original program is changed into a subroutine, which is called by the actor script. The names of the script and the subroutine are created via macros that append "_script" and "_subroutine" onto the base of the filename. For example: vconvolve.c contains the script "vconvolve_script" and the subroutine "vconvolve_subroutine".

2. The file I/O is changed to port I/O by replacing each call to "readimage" with a call to "mgk_get_image", and each call to "writeimage" with a call to "mgk_send_image". The mgk_get_image and mgk_send_image routines receive/propagate images from/to the correct port by referring to the forementioned filename to I/O port conversion tables.

The kactor script behaves exactly as the stand–alone program did. The argument parsing, input checking, and error handling facilities are left intact. The main differences are (1) instead of reading and writing files, the kactors receive and propagate image data using the Multigraph message passing mechanism. (2) The program is executed as an actor under the control of the Multigraph kernel instead of as a stand–alone program under the Unix shell.

The following khoros programs were successfully tested as kactors.

- vconvert (converts image data types)

- vconvolve (convolves an image with an arbitrary kernel)

- venhance (performs pixel re–mapping based on local statitistics)

- vnormal (normalizes an image)

- vsub (subtracts two images)

- vhstr (performs a histogram stretch)

- putimage (displays an image with update)

- vgauss (creates a gaussian image)

- vgcirc (creates the image of a filled circle)

- vrtoc (converts real data to complex)

- vbandcomb (combines bands from m images into one multiband image)

- vtranspos (transposes an image)

- vpad (pads a data sequence with a constant)

- vbandspt1 (extracts a specified data band from a multiband image)

- vfft (performs a 2-D fft on an image)

- vbrf (applies a band reject filter)

- vmoredge (performs a morphological edge extraction)

- vabs (computes the absolute value of an image)

The kactor conversion technique is applicable to any khoros program, so the new system can take advantage of the entire image processing library of Khoros.

The PPML Modeling Language:

PPML stands for Parallel Processing Modeling Language. It was developed at Vanderbilt University by Dr. Gabor Karsai [2]. PPML is a declarative language that includes the facilities to model three aspects of a processing network:

- The processing (data flow) aspect

- The hardware aspect

- The allocation of sub-tasks to the hardware.

The PPML models are made up definitions of the following types:

- primitive

- compound

- node

- network

- configuration

Primitive and compound definitions form the processing aspect model. Node and network declarations comprise the hardware aspect model. The configuration definition models the system as a whole, including the sub-task to processor allocation.

A primitive is the elemental processing block, and is described by a PPML "defprimitive" declaration. This declaration contains a primitive name, a triggering mode, an I/O interface description, a context description, and a script. Each of the primitive definitions represents a separate kind of process, or actor, that can be used in the signal flow. A compound is a group of primitive instances which are interconnected [1]. Compounds are declared by a "defcompound" block in PPML. A defcompound block contains a name, an I/O interface description, a list of signals (data nodes), and a structure definition. In the structure definition the internal connectivity of the compound is described, and the context of each primitive instance is set. The node and network models describe the hardware. A defnode declaration describes a particular type of hardware element (or node), such as a Sun sparc station or a TIC40 processor. The defnet declaration describes a hardware network which is made up of instances of the defined node types. The configuration model is represented textually in a defconfig declaration, which includes an instance of a compound (processing), an instance of a network (processors), and an allocation (a mapping of the processing blocks to the network nodes). A more complete description of PPML will be given in a user manual in the future.

An example PPML script is shown below. This script was created using the Khoros-PPML translator that is described in the next section. It represents a signal flow that was created in Cantata. Note that the first part, up until the defnode declaration, represents the signal flow. The signal flow is made up of 10 actors of 7 varieties, and 7 signals which connect them together. The defnode and defnet blocks define a hardware network made up of two "sparc" nodes, called "proc1" and "proc2". The defconfig declaration is a mapping

---

[1]Since PPML is a hierarchical language, the structure can also include instances of other compounds. The top-level compound is distinguishable by the fact that it will have no inputs or outputs in its I/O interface description.

of the primitive instances to the processing nodes in the network. Note that all of the actors in the graph are allocated to proc1. This is the default of the translator. At this time, the allocation of parallel tasks to hardware must be done manually.

```
;*********************************************************************

(defprimitive vconvert_type1 ifall ((voidp input1) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script " extern void vconvert_script(); vconvert_script(&command); "))

(defprimitive putimage_type1 ifall ((voidp input1) -> )
  ((string command) (string input_keys) (string output_keys))
  (script "extern void putimage_script(); putimage_script(&command); "))

(defprimitive vconvolve_type1 ifall ((voidp input1)(voidp input2) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script " extern void vconvolve_script(); vconvolve_script(&command); "))

(defprimitive vsub_type1 ifall ((voidp input1)(voidp input2) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script " extern void vsub_script(); vsub_script(&command); "))

(defprimitive vnormal_type1 ifall ((voidp input1) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script " extern void vnormal_script(); vnormal_script(&command); "))

(defprimitive readimage_type1 ifall ( -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script " extern void readimage_script(); readimage_script(&command); "))

(defprimitive writeimage_type1 ifall ((voidp input1) -> )
  ((string command) (string input_keys) (string output_keys))
  (script " extern void writeimage_script(); writeimage_script(&command); "))

(defcompound ultra (->) () ()
  (signals (voidp dn1) (voidp dn2) (voidp dn3) (voidp dn4)
          (voidp dn5) (voidp dn6) (voidp dn7))
  (struct
```

```
      (vconvert_type1_num1 (vconvert_type1 ( dn1  -> dn2)
        ("vconvert -i ./ultrasound.xv -o ./AAA -t \"float\" -b 0 " "-i" "-o")))
      (putimage_type1_num1 (putimage_type1 ( dn1  -> )
        ("putimage -i ./ultrasound.xv -update 2 " "-i" "")))
      (vconvolve_type1_num1 (vconvolve_type1 ( dn2  dn3  -> dn4)
        ("vconvolve -i1 ./AAA -i2 ./edgekern -o ./AAB " "-i1 -i2" "-o")))
      (vsub_type1_num1 (vsub_type1 ( dn2  dn4  -> dn5)
        ("vsub -i1 ./AAA -i2 ./AAB -o ./AAC " "-i1 -i2" "-o")))
      (vnormal_type1_num1 (vnormal_type1 ( dn5  -> dn6)
        ("vnormal -i ./AAC -o ./AAD -n 255 " "-i" "-o")))
      (vconvert_type1_num2 (vconvert_type1 ( dn6  -> dn7)
        ("vconvert -i ./AAD -o ./output.xv -t \"byte\" -b 0 " "-i" "-o")))
      (putimage_type1_num2 (putimage_type1 ( dn7  -> )
        ("putimage -i ./output.xv -update 2 " "-i" "")))
      (readimage_type1_num1 (readimage_type1 ( -> dn1)
        ("readimage -o ./ultrasound.xv" "" "-o")))
      (readimage_type1_num2 (readimage_type1 ( -> dn3)
        ("readimage -o ./edgekern" "" "-o")))
      (writeimage_type1_num1 (writeimage_type1 ( dn7  -> )
        ("writeimage -i ./output.xv" "-i" "")))
  ))


(defnode sparc ()
  (speed 1.0 0.001)
  (sizes (char   1 1) (uchar  1 1)
 (short  2 2) (ushort 2 2)
 (int    4 4) (uint   4 4)
 (float  4 4) (double 8 8)
 (string 4 4) (voidp  4 4))
  (queueing fcfs)
  (cc "gcc -g -c -I./ ~a")
  (ld "build-khoros ~a serv")
  (sh "(./serv 8000 &); sleep 1")
  (ident (values "alex" 0)))

(defnet proc ()
  (internals ethers)
  (init )
```

```
 (parts
  (proc1 (sparc ()))
  (proc2 (sparc ())))))

(defconfig test ()
  (processing (ultra (ultra ())))
  (processors (spa (proc ())))
  (allocation
   (ultra readimage_type1_num1  => spa proc1)
   (ultra readimage_type1_num2  => spa proc1)
   (ultra writeimage_type1_num1 => spa proc1)
   (ultra vconvert_type1_num1   => spa proc1)
   (ultra vconvert_type1_num2   => spa proc1)
   (ultra vconvolve_type1_num1  => spa proc1)
   (ultra vsub_type1_num1       => spa proc1)
   (ultra vnormal_type1_num1    => spa proc1)
   (ultra putimage_type1_num1   => spa proc1)
   (ultra putimage_type1_num2   => spa proc1)
   ))


;*********************************************************
```

The Model Level Khoros–Multigraph Interface:

The model level Khoros–Multigraph iterface consists of a model translator that creates
a PPML declaration from a Cantata data–flow model. The Cantata data–flow model is
obtained by capturing the standard output of Cantata as the graph is executed, which
consists of the exact commands that the user would type if he (or she) were executing the
data–flow step by step from the command line. This information is adequate to reconstruct
the signal flow of the system, since it contains the program names, the algorithmic parame-
ters and flags, and the filenames (the filenames specify the interconnectivity of the graph).
The translator is a stand–alone C program which reads in the logged standard output from
Cantata and writes the corresponding PPML declaration. An example of the operation of
the translator will be given in the next section.

Results:

The Khoros–Multigraph image processing system was tested for a number of different data–flows. Each of the khoros functions mentioned above were tested in at least one graph. An example is given below that demonstrates the operation of the system. Note first the Cantata data–flow shown in figure3. This graph, which is an example released with the standard distribution of Khoros[2], perfoms an edge detection on an artificially generated an image and displays the result. As the graph in figure3 was executed, the standard output

Figure 3: Example Cantata Data–Flow

of Cantata was captured and piped to a log file, which was then translated into a PPML declaration. Both the log file and the PPML models are shown below.

---

[2]This workspace can be found in KHOROS_HOME/repos/workspace/vipl_wksp/vipl.Z

```
;*********************************************************************
;                 This is the standard out from Cantata
;*********************************************************************


uncompress -c ../khoros/repos/workspace/vipl_wksp/vipl.Z
vgauss -o ./vgausAAAa001ss -c 150 -r 128 -x 64 -y 64 -vx 30 -vy 30 -ra 0 -amp 1 -n 0
vgcirc -o ./vgcirCAAa001ss -d 50 -x 64 -y 64 -t "byte" -b 0 -f 255 -r 128 -c 150
vrtoc -i1 ./vgausAAAa001ss -o ./vrtocBAAa001ss -t 1
vconvert -i ./vrtocBAAa001ss -o ./vconvDAAa001ss -n 255 -t "byte" -b 0
vbandcomb -i1 ./vgcirCAAa001ss -i2 ./vconvDAAa001ss -o ./vbandLAAa001ss -m 0 -c 0
vtranspos -i ./vbandLAAa001ss -o ./vtranMAAa001ss
vpad -i ./vtranMAAa001ss -o ./vpadXNAAa001ss -r 256 -c 256 -d 0 -e 0 -j 0 -k 0
vbandspt1 -i ./vpadXNAAa001ss -o ./vbandOAAa001ss -b 0
vhstr -i ./vbandOAAa001ss -o ./vhstrEAAa001ss -p 0
vconvert -i ./vhstrEAAa001ss -o ./vconvGAAa001ss -n 255 -t "float" -b 0
vfft -i1 ./vconvGAAa001ss -o1 ./vfftFFAAa001ss -d 0
vbrf -i ./vfftFFAAa001ss -o ./vbrfMHAAa001ss -n 1 -c 0.5 -w 10
vfft -i1 ./vbrfMHAAa001ss -o1 ./vfftNIAAa001ss -d 1
vconvert -i ./vfftNIAAa001ss -o ./vconvKAAa001ss -n 255 -t "byte" -b 0
vmoredge -i ./vconvKAAa001ss -k ./cell20.mor -o ./vmoreJAAa001ss
putimage -i ./vmoreJAAa001ss -x 700 -y 0 -update 2


;*********************************************************************
;               This is the translated PPML declaration
;*********************************************************************


(defprimitive vgauss_type1 ifall ( -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script "vgauss_script"))

(defprimitive vgcirc_type1 ifall ( -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script "vgcirc_script"))

(defprimitive vrtoc_type1 ifall ((voidp input1) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script "vrtoc_script"))
```

6-15

```
(defprimitive vconvert_type1 ifall ((voidp input1) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script "vconvert_script"))


(defprimitive vbandcomb_type1 ifall ((voidp input1)(voidp input2) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script "vbandcomb_script"))


(defprimitive vtranspos_type1 ifall ((voidp input1) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script "vtranspos_script"))


(defprimitive vpad_type1 ifall ((voidp input1) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script "vpad_script"))


(defprimitive vbandspt1_type1 ifall ((voidp input1) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script "vbandspt1_script"))


(defprimitive vhstr_type1 ifall ((voidp input1) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script "vhstr_script"))


(defprimitive vfft_type1 ifall ((voidp input1) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script "vfft_script"))


(defprimitive vbrf_type1 ifall ((voidp input1) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script "vbrf_script"))


(defprimitive vmoredge_type1 ifall ((voidp input1)(voidp input2) -> (voidp output1))
  ((string command) (string input_keys) (string output_keys))
  (script "vmoredge_script"))


(defprimitive putimage_type1 ifall ((voidp input1) -> )
  ((string command) (string input_keys) (string output_keys))
```

```
      (script "putimage_script"))


  (defprimitive readimage_type1 ifall ( -> (voidp output1))
    ((string command) (string input_keys) (string output_keys))
    (script "readimage_script"))


  (defprimitive writeimage_type1 ifall ((voidp input1) -> )
    ((string command) (string input_keys) (string output_keys))
    (script "writeimage_script"))


  (defcompound test (->) () ()
    (signals (voidp dn1) (voidp dn2) (voidp dn3) (voidp dn4)
             (voidp dn5) (voidp dn6) (voidp dn7) (voidp dn8)
             (voidp dn9) (voidp dn10) (voidp dn11) (voidp dn12)
             (voidp dn13) (voidp dn14) (voidp dn15) (voidp dn16))
    (struct
      (vgauss_type1_num1 (vgauss_type1 ( -> dn1)
        ("vgauss -o ./vgausAAAa001ss -c 150 -r 128 -x 64 -y 64 -vx 30
                                        -vy 30 -ra 0 -amp 1 -n 0 " "" "-o")))
      (vgcirc_type1_num1 (vgcirc_type1 ( -> dn2)
        ("vgcirc -o ./vgcirCAAa001ss -d 50 -x 64 -y 64 -t byte -b 0
                                        -f 255 -r 128 -c 150 " "" "-o")))
      (vrtoc_type1_num1 (vrtoc_type1 ( dn1  -> dn3)
        ("vrtoc -i1 ./vgausAAAa001ss -o ./vrtocBAAa001ss -t 1 " "-i1" "-o")))
      (vconvert_type1_num1 (vconvert_type1 ( dn3  -> dn4)
        ("vconvert -i ./vrtocBAAa001ss -o ./vconvDAAa001ss -n 255
                                        -t byte -b 0 " "-i" "-o")))
      (vbandcomb_type1_num1 (vbandcomb_type1 ( dn2  dn4  -> dn5)
        ("vbandcomb -i1 ./vgcirCAAa001ss -i2 ./vconvDAAa001ss -o
                          ./vbandLAAa001ss -m 0 -c 0 " "-i1 -i2" "-o")))
      (vtranspos_type1_num1 (vtranspos_type1 ( dn5  -> dn6)
        ("vtranspos -i ./vbandLAAa001ss -o ./vtranMAAa001ss " "-i" "-o")))
      (vpad_type1_num1 (vpad_type1 ( dn6  -> dn7)
        ("vpad -i ./vtranMAAa001ss -o ./vpadXNAAa001ss -r 256 -c 256
                                        -d 0 -e 0 -j 0 -k 0 " "-i" "-o")))
      (vbandspt1_type1_num1 (vbandspt1_type1 ( dn7  -> dn8)
        ("vbandspt1 -i ./vpadXNAAa001ss -o ./vbandOAAa001ss -b 0 " "-i" "-o")))
      (vhstr_type1_num1 (vhstr_type1 ( dn8  -> dn9)
```

```
            ("vhstr -i ./vbandOAAa001ss -o ./vhstrEAAa001ss -p 0 " "-i" "-o")))
        (vconvert_type1_num2 (vconvert_type1 ( dn9  -> dn10)
          ("vconvert -i ./vhstrEAAa001ss -o ./vconvGAAa001ss -n 255
                                            -t float -b 0 " "-i" "-o")))
        (vfft_type1_num1 (vfft_type1 ( dn10  -> dn11)
          ("vfft -i1 ./vconvGAAa001ss -o1 ./vfftFFAAa001ss -d 0 " "-i1" "-o1")))
        (vbrf_type1_num1 (vbrf_type1 ( dn11  -> dn12)
          ("vbrf -i ./vfftFFAAa001ss -o ./vbrfMHAAa001ss -n 1 -c 0.5
                                            -w 10 " "-i" "-o")))
        (vfft_type1_num2 (vfft_type1 ( dn12  -> dn13)
          ("vfft -i1 ./vbrfMHAAa001ss -o1 ./vfftNIAAa001ss -d 1 " "-i1" "-o1")))
        (vconvert_type1_num3 (vconvert_type1 ( dn13  -> dn14)
          ("vconvert -i ./vfftNIAAa001ss -o ./vconvKAAa001ss -n 255 -t byte
                                            -b 0 " "-i" "-o")))
        (vmoredge_type1_num1 (vmoredge_type1 ( dn14  dn15  -> dn16)
          ("vmoredge -i ./vconvKAAa001ss -k ./cell20.mor -o ./vmoreJAAa001ss "
                                            "-i -k" "-o")))
        (putimage_type1_num1 (putimage_type1 ( dn16  -> )
          ("putimage -i ./vmoreJAAa001ss -x 700 -y 0 -update 2 " "-i" "")))
        (readimage_type1_num1 (readimage_type1 ( -> dn15)
          ("readimage -o ./cell20.mor" "" "-o")))
        (writeimage_type1_num1 (writeimage_type1 ( dn16  -> )
          ("writeimage -i ./vmoreJAAa001ss" "-i" "")))
    ))


(defconfig test_config ()
  (processing (test (test ())))
  (processors (spa (proc ())))
  (allocation
    (test vgauss_type1_num1 => spa proc1)
    (test vgcirc_type1_num1 => spa proc1)
    (test vrtoc_type1_num1 => spa proc1)
    (test vconvert_type1_num1 => spa proc1)
    (test vbandcomb_type1_num1 => spa proc1)
    (test vtranspos_type1_num1 => spa proc1)
    (test vpad_type1_num1 => spa proc1)
    (test vbandspt1_type1_num1 => spa proc1)
    (test vhstr_type1_num1 => spa proc1)
```

```
(test vconvert_type1_num2 => spa proc1)
(test vfft_type1_num1 => spa proc1)
(test vbrf_type1_num1 => spa proc1)
(test vfft_type1_num2 => spa proc1)
(test vconvert_type1_num3 => spa proc1)
(test vmoredge_type1_num1 => spa proc1)
(test putimage_type1_num1 => spa proc1)
(test readimage_type1_num1 => spa proc1)
(test writeimage_type1_num1 => spa proc1)
))


;************************************************************
```

The PPML models shown above were loaded into the model interpreter and the graph was executed by the Khoros–Multigraph run–time system. Note that each of the Khoros programs used in this Cantata workspace had previously been converted to Kactors and linked with the Multigraph kernel. The resulting output from this execution was identical to the output of the Cantata execution. The execution times were also very similar. On a Sun Sparc10 running Solaris 5.2, the Cantata version took 31 seconds, and the Multigraph version took 29 seconds. This speedup is minimal, as expected. Only one processor was used in each case.

By using Multigraph, we have gained the ability to easily parallelize the data–flow. By altering the allocation section of the configuration model, any actor of the graph can be executed on any processor in the network. Thus, it would be possible to distribute the example graph over a network of 10 processors without changing the result. It is important to note that this type of parallelization will not decrease the execution time of a graph unless the graph has a naturally concurrent structure. A serial data–flow will remain serial even if the sub–tasks are executed on different nodes.

In order to guarantee an execution time speedup, a different approach is necessary. The image can be split into "p" pieces, each of which can be processed concurrently by a separate processor. The processed pieces can then be merged back together to form the output. This technique is called "split–and–merge" parallelization, and has been shown to be effective for reducing the execution times of many of the commonly used image processing algorithms [4] [5] [6] [6]. The split–and–merge paradigm can easily be added to the present system, which will greatly enhance its performance.

The new system will work as follows: The user will first build and test a data–flow in Cantata, as before. As an added step to the modeling process, he (or she) will designate split–and–merge processing blocks. The translator will then create a data parallel PPML data–flow model, leaving the number of parallel channels as an unspecified variable. During interpretation, the model interpreter will determine the number of parallel channels based on relavent factors (see [6]), and will automatically allocate the parallel channels to the available hardware. The execution time (latency) of the resulting split–and–merge data–flow graph will be much less than the execution time of the corresponding Cantata data–flow. Thus, with the addition of the split–and–merge processing paradigm, it is expected that the Khoros–Multigraph image processing system will greatly out–perform Khoros, but will be every much as easy to use.

RECOMMENDATIONS FOR FUTURE RESEARCH:

Although the Khoros–Multigraph system was successfully implemented, the real goals of its implementation have not yet been met. The execution time under the Multigraph network was marginally better while processing on one node. However, the goal is a high–performance system. The desired performance can be attained with the use of the split–and–merge processing paradigm, as has been shown by previous results [4] [5] [6] [6]. The split–and–merge processing technique was not include merely due to the limited time span of the RDL summer research program. In the near future, facilities should be added to the system to include the this feature.

ACKNOWLEDGEMENTS:

References:

[1] Biegl, Csaba.: "Multigraph Kernel (MGK) User's Manual", Dept. of Electrical Engineering, Vanderbilt University, 1988.

[2] Karsai, Gabor: "A Coordination Language For Parallel Processing", Department of Electrical Engineering, Vanderbile University, Aug 30, 1993.

[3] The Khoros Group: "Khoros Manual", Volume II, "Programmer's Manual", Department of Electrical and Computer Engineering, University of New Mexico, May 8, 1991.

[4] Moore, Michael S.: "Multigraph Implementation of Image Morphology", Final report for the 1991 USAF-RDL summer research program, 1991.

[5] Moore, Michael S.: "A Multigraph Implementation of a Distributed Image Processing System", Final report for the 1992 USAF-RDL summer research program, August 1992.

[6] Moore, M.S.: "A Programming Environment for Parallel Image Processing: Progress Report," Report Prepared for Sverdrup Technology, Inc., 31 January 1993.

[7] Moore, M.S.: "A Programming Environment for Parallel Image Processing: Progress Report 2," Report Prepared for Sverdrup Technology, Inc., 31 March, 1993.

[8] Souder, Richard S.: "Parallel Distributed Image Processing", Master's Thesis Submitted to the Faculty of the Graduate School of Vanderbilt University, 1989.

[9] Sztipanovits, J., Abbott, B., and Biegl, C.: "Programming Environment for Parallel Image Processing," Report Prepared for Sverdrup Technology, Inc., September, 1992

Initial Evaluation of Adaptive Engine Simulation Model

Kimball A. Shahrokhi

Department of Mechanical Engineering
Vanderbilt University
Nashville, TN 37235

Initial Evaluation of
Adaptive Engine
Simulation Model

Kimball A. Shahrokhi

Department of Mechanical Engineering
Vanderbilt University

Abstract

The concept of system identification was discussed, as well as a specific identification program called SCIDNT (Systems Control Identification). An engine simulation program was also reviewed, specifically ATEST (Advanced Turbine Engine Simulation Technique). An adaptive simulation routine, whose origins were from a mainframe operating environment, was modified for use on a DOS operating system. The adaptive engine simulation routine was then evaluated using turbine engine test data from an engine test cell. It was concluded from the evaluation that an engine model trimmed using a system identification code results in an overall improvement in the prediction of engine measurements.

Introduction

Over the past several years, engineers at Arnold Engineering Development Center (AEDC) have been developing mathematical models to simulate various aircraft systems, including turbine engines. A primary use of these models is to examine engine response in operating regimes that are difficult or costly to examine in the engine test cell. Engineers are constantly establishing and evaluating new codes, as well as "tuning" existing codes to accurately reflect the specific component and overall engine performance.

System identification is a technique that aids in the evaluation and "tuning" of mathematical models. System identification computer programs use a specified model and a given set of data, and then attempt to optimize the model to accurately reflect the data. The result is a set of parameters or "scale factors" which adjusts the model to reflect the data, or rejects the model as unfit to represent the physical system.

The use of a system identification technique in identifying parameters within a turbine engine simulation model is explored within this report. Based upon actual engine data, the identification routine combined with the engine model is referred to as an adaptive engine simulation, due to the ability of the model to adjust to the current state of the engine.

The following report begins with a background of the system identification code used, as well as a brief discussion of the mathematical model employed. Next, the objectives are defined and the approach taken to achieve the objectives stated. Finally, the results are presented, followed by recommendations for future work.

Background

The system identification routine that is used is known as SCIDNT (Systems Control Identification), originally developed by Systems Control Technology Inc., under contract from AEDC. SCIDNT identifies, or estimates, model parameters to fit the model to experimental data based upon criteria specified by the user.

The basics of system identification are examined in detail by Norton [1] and Eykhoff [2]. The system being modeled is an aircraft turbine engine. This type of system, when modeled, is referred to as a nonlinear, multiple-input-multiple-output system. Two common approaches to identifying the parameters of such a complex model are the equation error method and the output error method [3].

Both of the above mentioned methods estimate unknown parameters through the minimization of a performance index [3]. The equation error method, also called the least squares method, is effective at dealing with a dynamic process. However, this estimation method does not deal well with disturbances in the measurements (i.e. measurement noise). Conversely, the output error method deals well with measurement noise, but does not deal well with process dynamics or "process noise".

The SCIDNT identification routine utilizes an estimation method which combines the advantages of both methods. This method is a variation of the Gauss-Newton (GN) method known as the Levenberg-Marquardt (LM) algorithm. As with the equation error and output error methods, the LM algorithm within SCIDNT seeks to minimize a performance index. As previously mentioned, the LM algorithm is a variation of a GN method; the variation being that the algorithm is a combination of a gradient search and a GN method to search for a minimum.

The performance index that SCIDNT uses is a cost function associated with the error corresponding between the actual versus the simulated measurements. SCIDNT seeks to minimize the cost function through the modification of user specified model parameters. Model parameters corresponding to a minimized cost function are referred to as estimated or "optimized" parameters which produce a trimmed model.

An important feature of SCIDNT that must be addressed is that weighting factors are associated with the measurements. Each measurement has a corresponding weighting factor that must be specified by the user. A modification of a particular weighting factor increases or decreases the impact of the corresponding measurement error on the overall cost function. A drop in the weighting factor corresponds to an increase in the impact; a nominal weighting factor setting is 1.0. Representative weighting factors, as well as the parameters and measurements, are shown in Tables A1 and A2.

Setting all weighting factors to 1.0 does not imply that all measurements have the same importance in the determination of the overall cost function. With the current program design and nominal weighting factors, the thrust measurements have a dominant role in the determination of the cost function. This current design is simply a starting point and future modifications of the various weighting factors is anticipated.

The turbine engine model that was used is known as ATEST (Advanced Turbine Engine Simulation Technique). This simulation code is a 1-D, component-matching routine, capable of modeling arbitrary engine configurations. For the current research, a dual-spool, mixed-flow turbofan engine model is employed, illustrated in Figure 1. The engine model can simulate both steady-state and transient operation (0 - 20 Hz), but not

dynamic operation. A more detailed description of ATEST is given by Chappell [4].
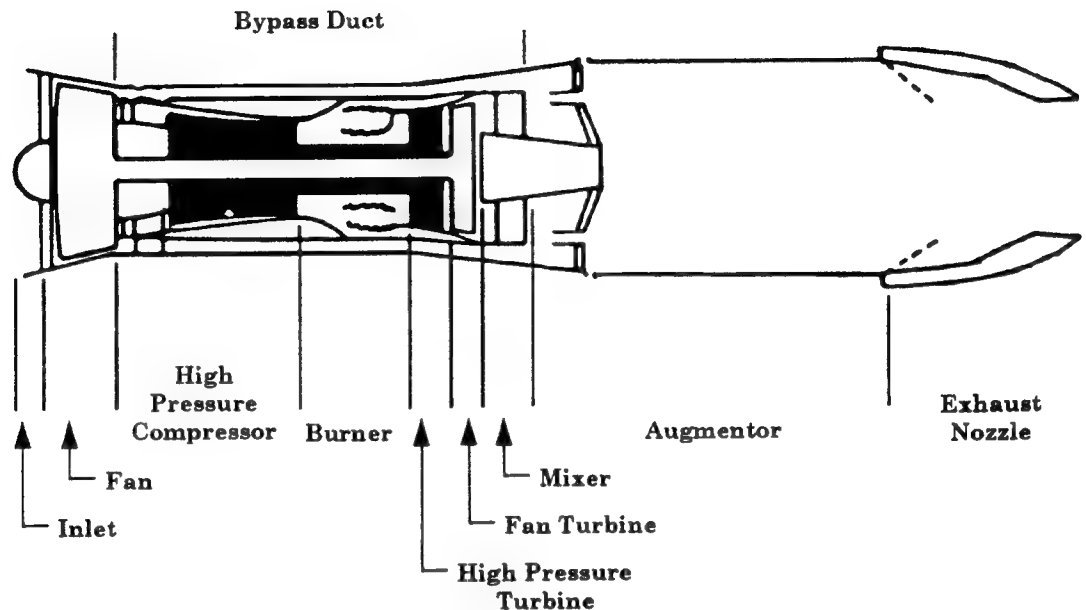

Figure 1. Schematic of a dual-spool, mixed-flow turbofan engine.

As a stand-alone program, ATEST will calculate a design point using user supplied input data. This design point establishes the scale factors for component performance maps and mixing plane areas. With a design point calculated, the model is in effect trimmed to a family of engines. Due to variations in engine characteristics (e.g. engine manufacturing, tolerances variations, and engine deterioration), the model output can not be expected to reflect exactly the performance for any particular engine.

Objectives

The first objective was to convert the ATEST and SCIDNT FORTRAN programs from a mainframe operating environment to a DOS operating environment. The second objective was to modify the adaptive simulation code to establish a means of evaluating the code. The last objective was to perform a limited evaluation of the adaptive simulation code.

Approach

The ATEST model and the identification program SCIDNT were converted from a mainframe to a DOS environment. For discussion purposes, the combination of SCIDNT and ATEST, shown if Figure 2, will be referred to as the adaptive simulation process. Newly integrated code is shown in the figure with a single line border.
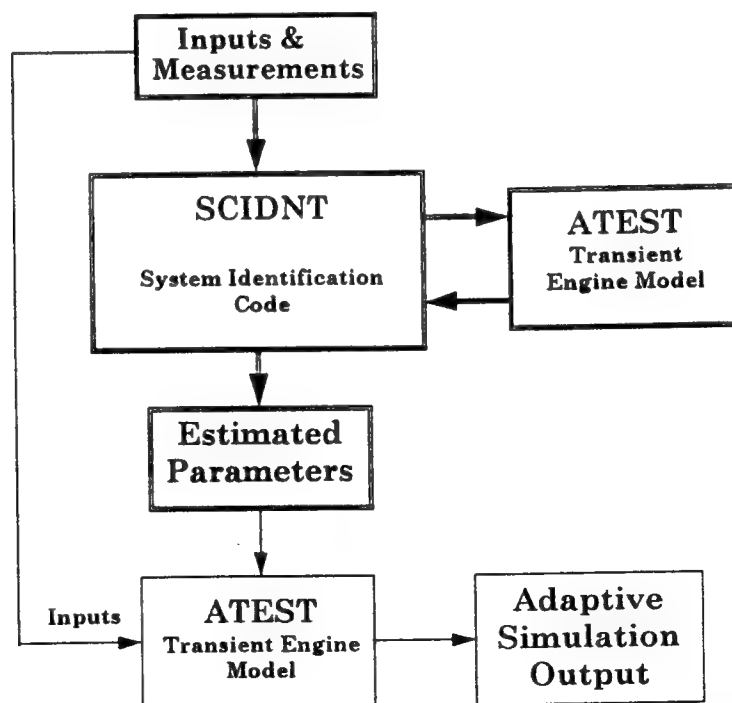
Figure 2. Flow chart of the adaptive simulation process.

Previous versions of the ATEST program had been converted to a DOS environment, thus this conversion was used as a learning process for the more formidable task of conversion of the SCIDNT program. Due to the standardization of the FORTRAN programming language, the bulk of the effort in the conversion process was spent on tailoring the I/O (i.e. input/output) functions to the conventions of the particular compiler, in this case, the Lahey FORTRAN compiler. One major modification came in the method of transfer of the test data to the SCIDNT program. In the mainframe environment, the test data was transferred to the program in binary form. It was decided that in the DOS environment that the test data would be transferred in ASCII form, which allows easier data manipulation for the program user.

Once the adaptive simulation code was operational, a modification was made to evaluate the performance of the code. The modification came in the form of an additional execution of the ATEST engine model with the estimated parameters from SCIDNT and the input test data. The resulting output from the additional execution are calculated measurements which can be compared to the actual test measurements.

Finally, a limited evaluation was performed on the adaptive simulation program. This evaluation compared the actual measurements to the simulated measurements from the trimmed, engine simulation model. These comparisons are presented and discussed in the results and conclusions sections.

Results

A limited evaluation on the adaptive simulation code was performed. Data from a particular engine was used and comparisons were made between the trimmed model output and the actual measurements. Data from the dual-spool, mixed-flow, turbofan engine are shown in Figures A1 to A11.

The evaluation began with an initial execution of ATEST, in stand-alone form, to calculate design point operation. This initial execution of ATEST creates a baseline from which the identification code can be evaluated and represents an engine model trimmed to a family of engines, but not to the particular engine tested.

In Figures A12 & A13, a comparison of actual measurements versus the simulated measurements is shown. One can see that the measured and simulated net thrusts compare well, while the engine airflow measurements differ significantly. Assuming valid data, the reason for this difference is that the engine model is not trimmed to the particular engine from which the data originated.

The next step in the evaluation process is to examine the change in simulated measurements when SCIDNT determines the model parameters. This initial identification is the first step in creating an engine simulation trimmed to simulate a particular engine with its individual characteristics. Figures A14 & A15 show the comparison of the actual measurements with those simulated using the adaptive engine simulation model. For this comparison, the weighting factors are at there nominal settings. SCIDNT output from this execution of the adaptive engine simulation is shown in Table A1..

As shown in Figure A14, the simulated net thrust compares favorably with the actual measurements. This is due to the dominance of the thrust

measurements on the determination of the cost function using nominal weighting factors. The simulated engine airflow resulted in an improvement over the baseline, but an appreciable difference still exists.

The next step in the evaluation was to examine the impact of modifying the weighting factors. Due to the limited evaluation, modification of only the airflow weighting factor could be evaluated. The modification in the weighting factor increased the importance of the airflow in the calculation of the overall cost function. In this execution, the airflow measurement is now on the same order of magnitude as the thrust measurements. Table A2 shows the SCIDNT output with the modified airflow weighting factor.

The results of the modified airflow weighting factor show a significant improvement in the simulated engine airflow. This improvement is due to the fact that the increased importance of the airflow measurement "drove" the simulated values closer to the actual, as seen in Figure A17. However, agreement between simulated and measured net thrust was worse (Figure A16). The disagreement can be attributed to, in part, the decrease in importance of the thrust measurements in the determination of the overall cost function.

Conclusions and Recommendations

The ATEST design point calculation improved the engine model simulation. The use of the SCIDNT identification code resulted in better agreement between the model and the actual engine tested. All the weighting factors must be modified to trim the engine model. Modification of the airflow weighting factor resulted in closer agreement with the airflow measurement. However, it is also apparent from the decreased agreement in

net thrust, that all the weighting factors must be examined and modified to produce the desired simulation accuracy.

Further examination including an in depth analysis of measurements and associated measurement uncertainties of the adaptive engine simulation is necessary to realize the full potential of this code. Initial evaluation of the system identification technique has proved it to be a powerful tool in tuning engine simulation mathematical models to more accurately predict as-tested engine performance.

References

1. Norton, J.P., An Introduction to Identification, Academic Press, Inc., Orlando, FL, 1986.

2. Eykhoff, P., System Identification, Parameter and State Estimation, John Wiley & Sons, London, 1974.

3. Anex, R. P., Application of System Identification Techniques to Turbine Engine Post-Stall Test and Evaluation, Arnold Engineering Development Center, Technical Report #90-32, vol. 2, December, 1990.

4. Chappell, M. and McLaughlin, P., "An Approach to Modeling Continuous Turbine Engine Operation from Startup to Shutdown", AIAA Paper No. 91-2373, Presented at the AIAA/SAE/ASME/ASEE Twenty-seventh Joint Propulsion Conference, June 24-26, 1991, Sacramento, CA.
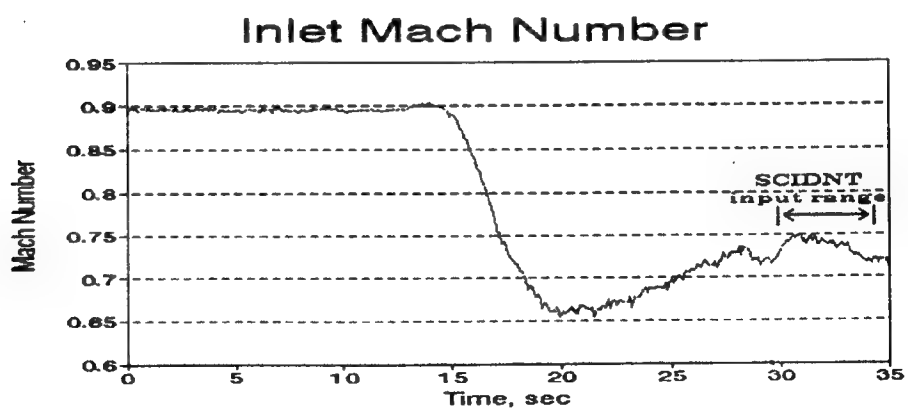
FINAL VALUES

FUNCTION VALUE= 1829604.4

MEASUREMENT	ERROR	RMS	FUNCTION BUDGET	WEIGHT	
FG		140.454	986372.	1.00000	
FN		172.014	0.147943E+07	1.00000	
FRAM		142.555	0.101609E+07	1.00000	
P13		1.37830	94.9860	1.00000	
P25		1.11314	61.9539	1.00000	
P3		13.0021	8452.78	1.00000	
T13		30.9582	47920.6	1.00000	
T25		25.4611	32413.5	1.00000	
T3		41.5665	86388.7	1.00000	
W2		6.28874	1977.41	1.00000	

PARAMETER	VALUE
1 WAC2SCA	9.6757847E-01
2 EFF2SCA	1.0217005E+00
3 PR2SCA	1.0135444E+00
4 XNC2SCA	8.7436050E-01
5 WAC25SCA	1.0432723E+00
6 EFF25SCA	9.5254391E-01
7 PR25SCA	9.9713647E-01
8 XNC25SCA	8.4555548E-01
15 AMIX16	5.0647099E+01
16 AMIX56	7.1390942E+02

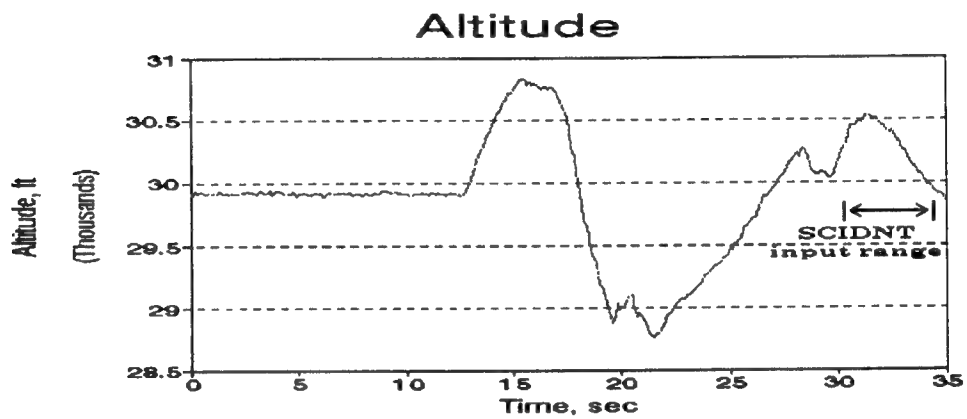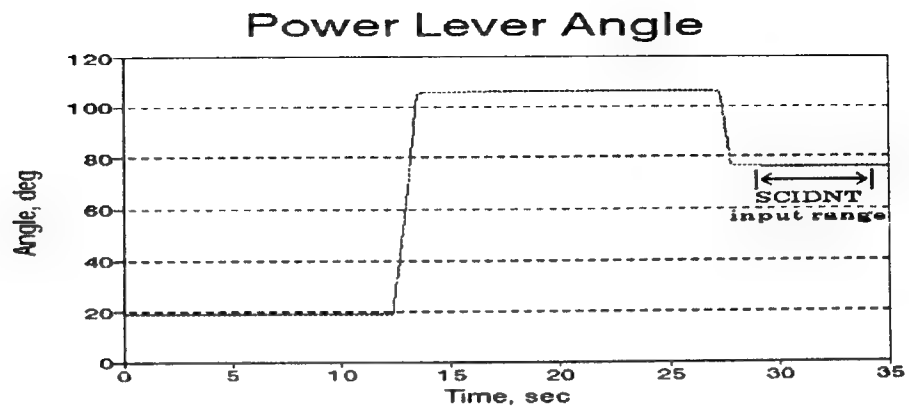Table A1. SCIDNT output with nominal weighting factors. (See reference 4 for nomenclature)
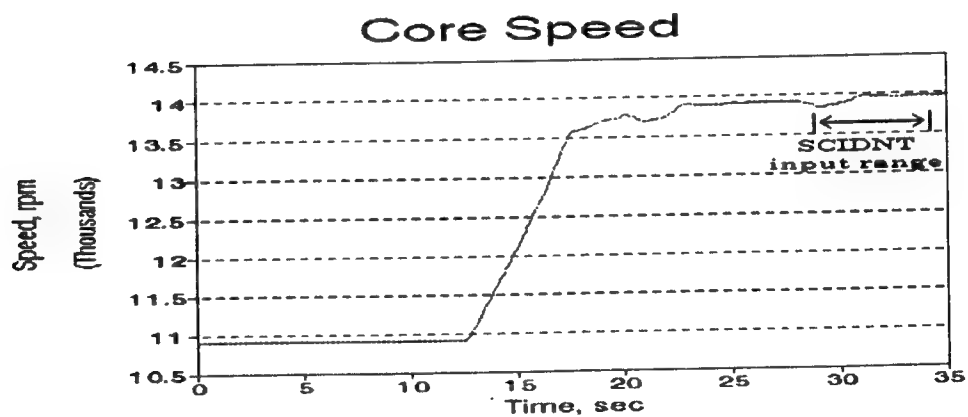
FINAL VALUES

FUNCTION VALUE= 4175071.6

MEASUREMENT	ERROR	RMS	FUNCTION BUDGET	WEIGHT
FG		207.230	0.214721E+07	1.00000
FN		245.647	0.301713E+07	1.00000
FRAM		51.0057	130079.	1.00000
P13		1.24736	77.7953	1.00000
P25		1.00294	50.2948	1.00000
P3		23.6521	27971.1	1.00000
T13		45.4212	103154.	1.00000
T25		41.0924	84429.4	1.00000
T3		76.1612	290027.	1.00000
W2		2.25832	0.255001E+07	0.100000E-01

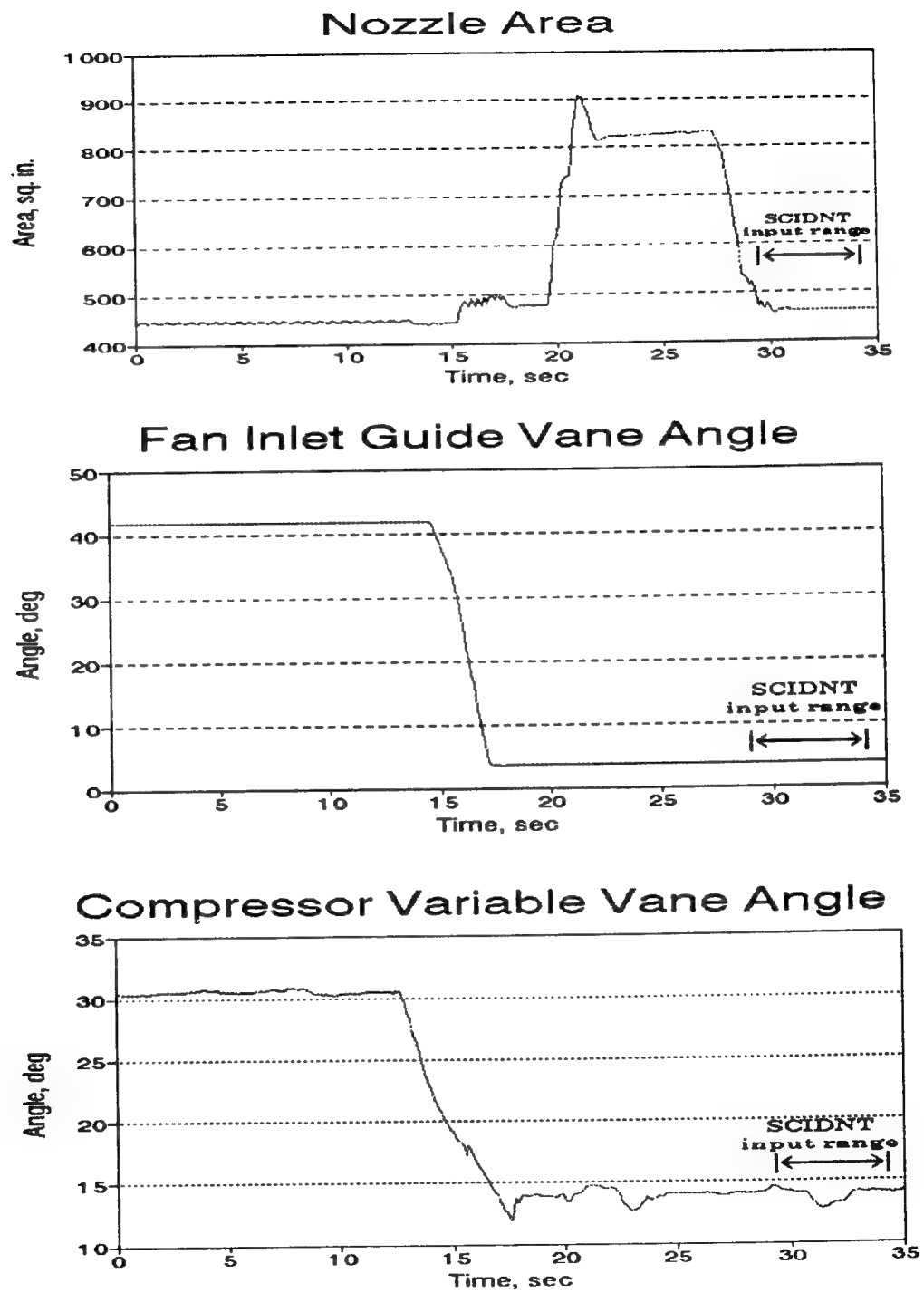| PARAMETER	VALUE
1 WAC2SCA	9.1950560E-01
2 EFF2SCA	1.0175272E+00
3 PR2SCA	1.0154516E+00
4 XNC2SCA	9.8229617E-01
5 WAC25SCA	1.0438070E+00
6 EFF25SCA	9.2382407E-01
7 PR25SCA	1.1610364E+00
8 XNC25SCA	6.1839682E-01
15 AMIX16	2.0432941E+01
16 AMIX56	3.0939557E+02

Table A2. SCIDNT output with modified airflow weighting factor.(See reference 4 for nomenclature)
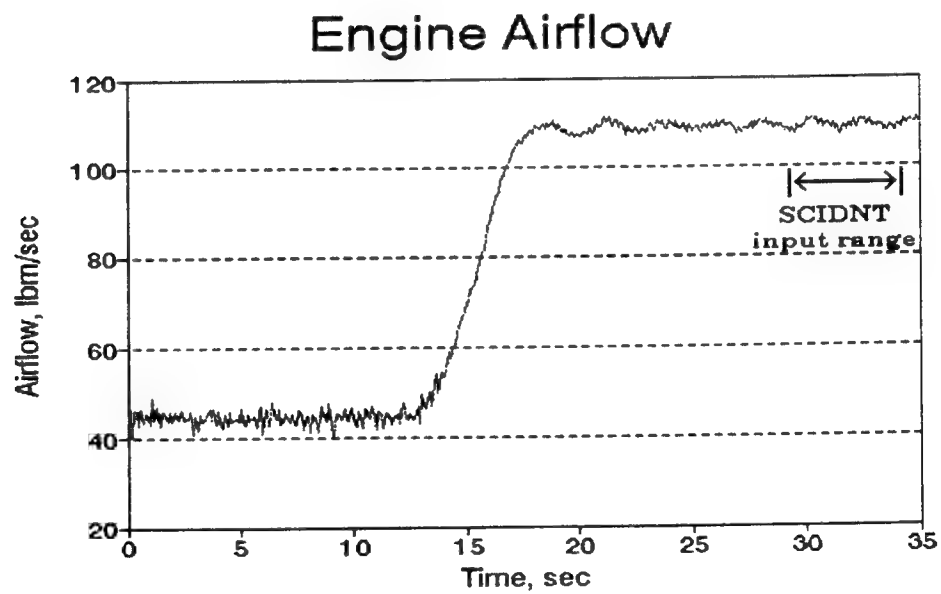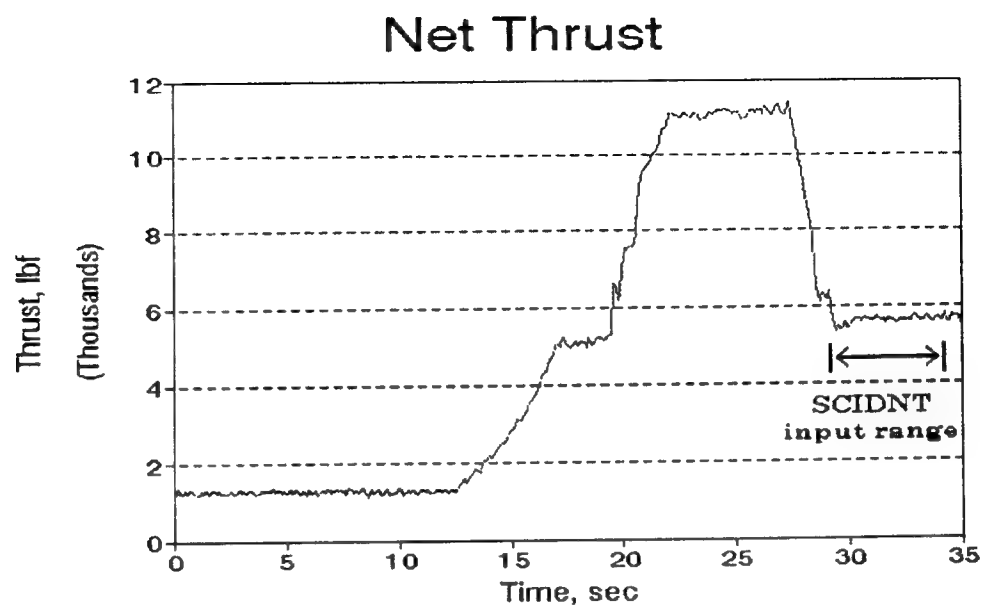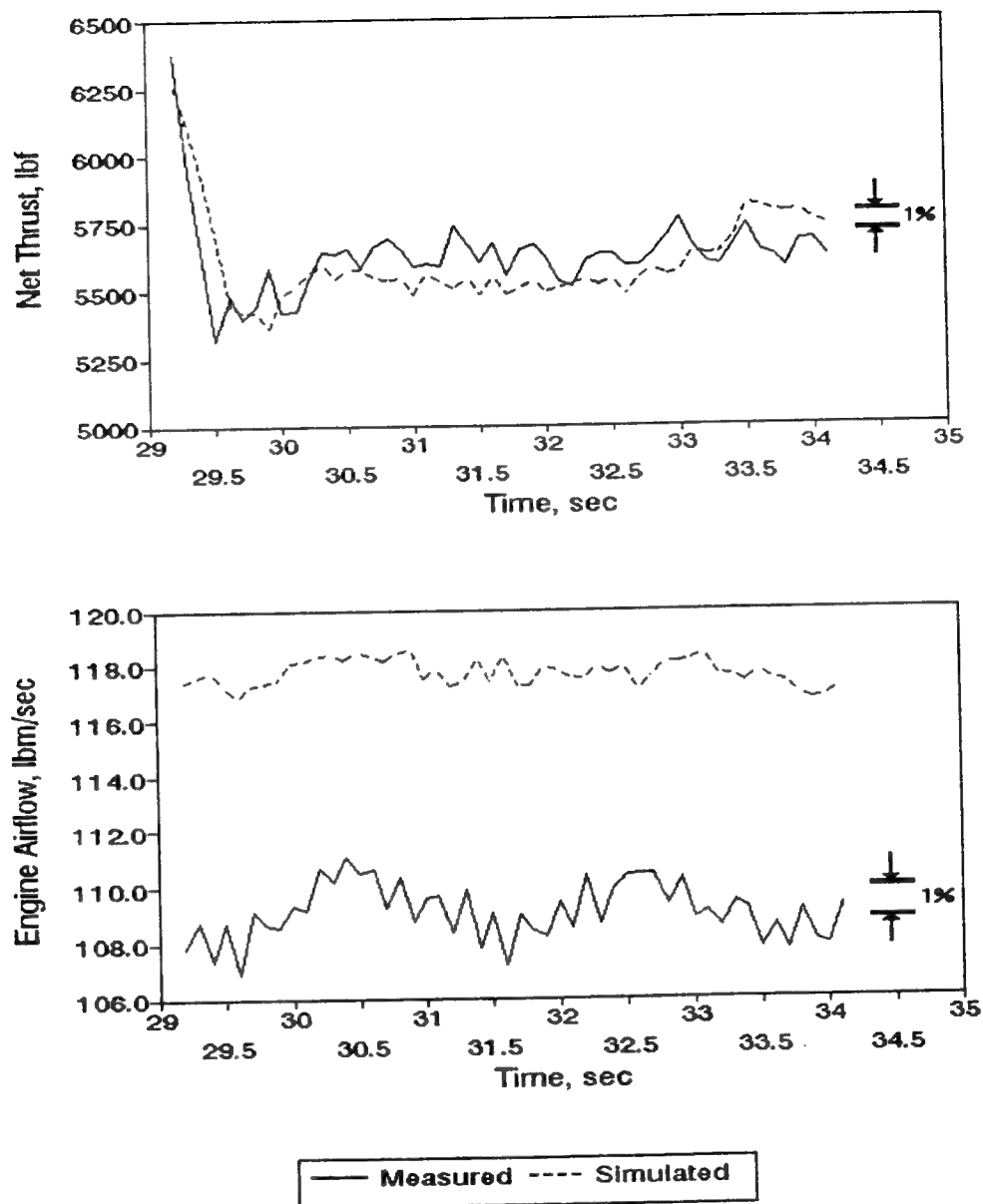
Figures A1, A2, & A3. Engine test data.

Total Engine Fuel Flow

Fan Speed

Core Speed
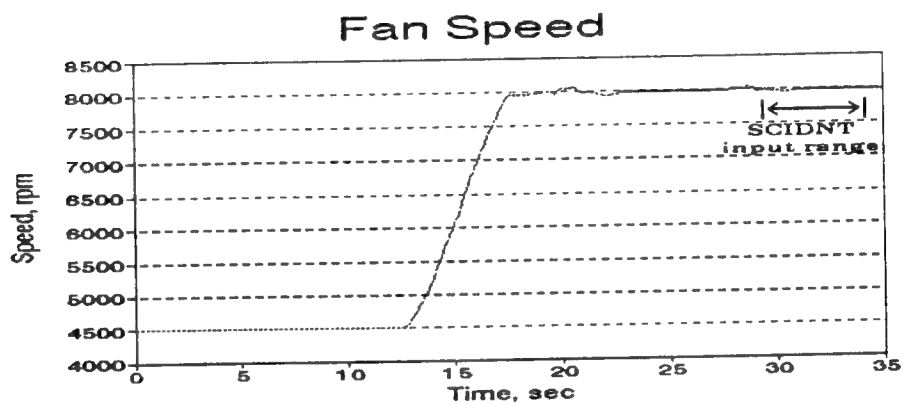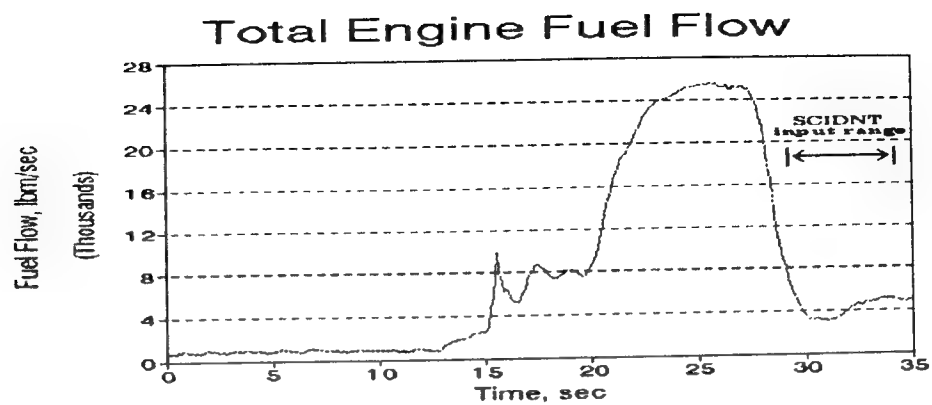
Figures A4, A5, & A6. Engine test data.

Nozzle Area

Area, sq. in. vs Time, sec

SCIDNT
input range

Fan Inlet Guide Vane Angle

Angle, deg vs Time, sec

SCIDNT
input range

Compressor Variable Vane Angle

Angle, deg vs Time, sec

SCIDNT
input range

Figures A7, A8, & A9. Engine test data.

Net Thrust
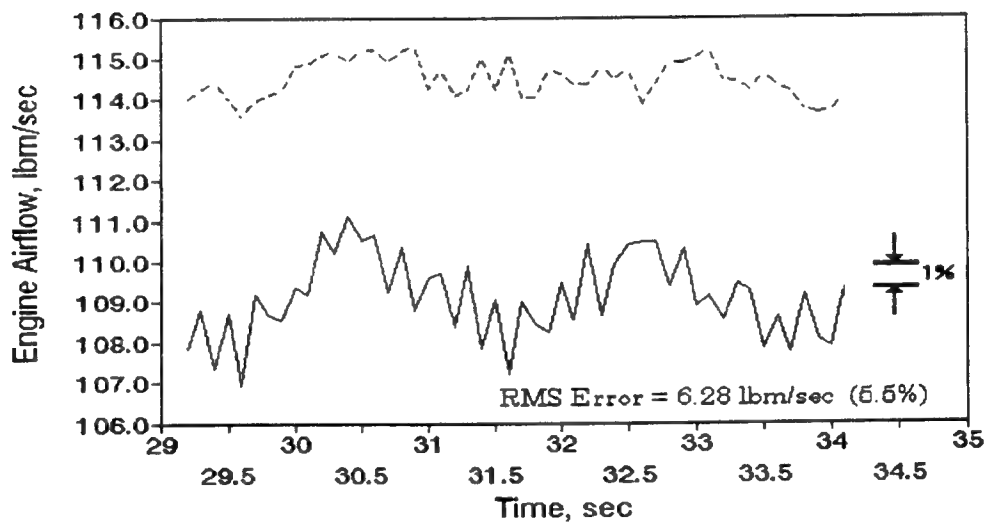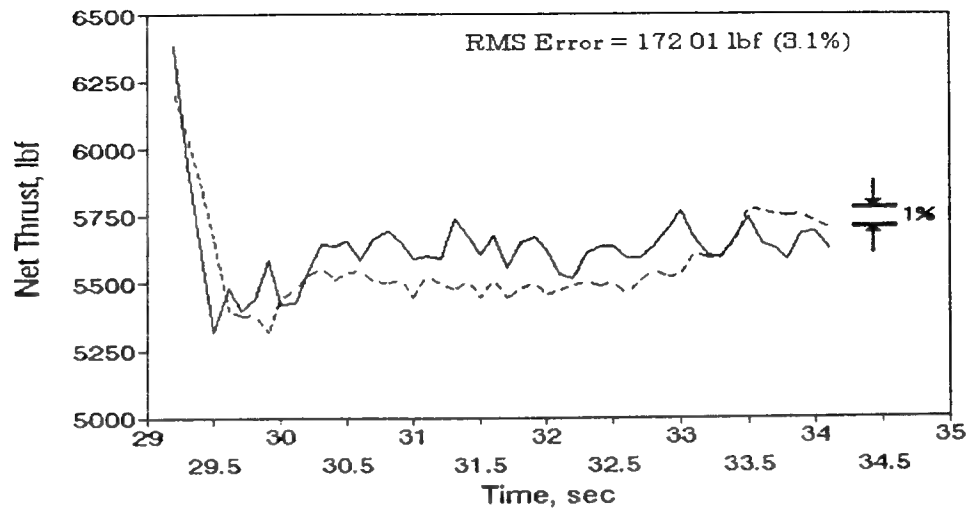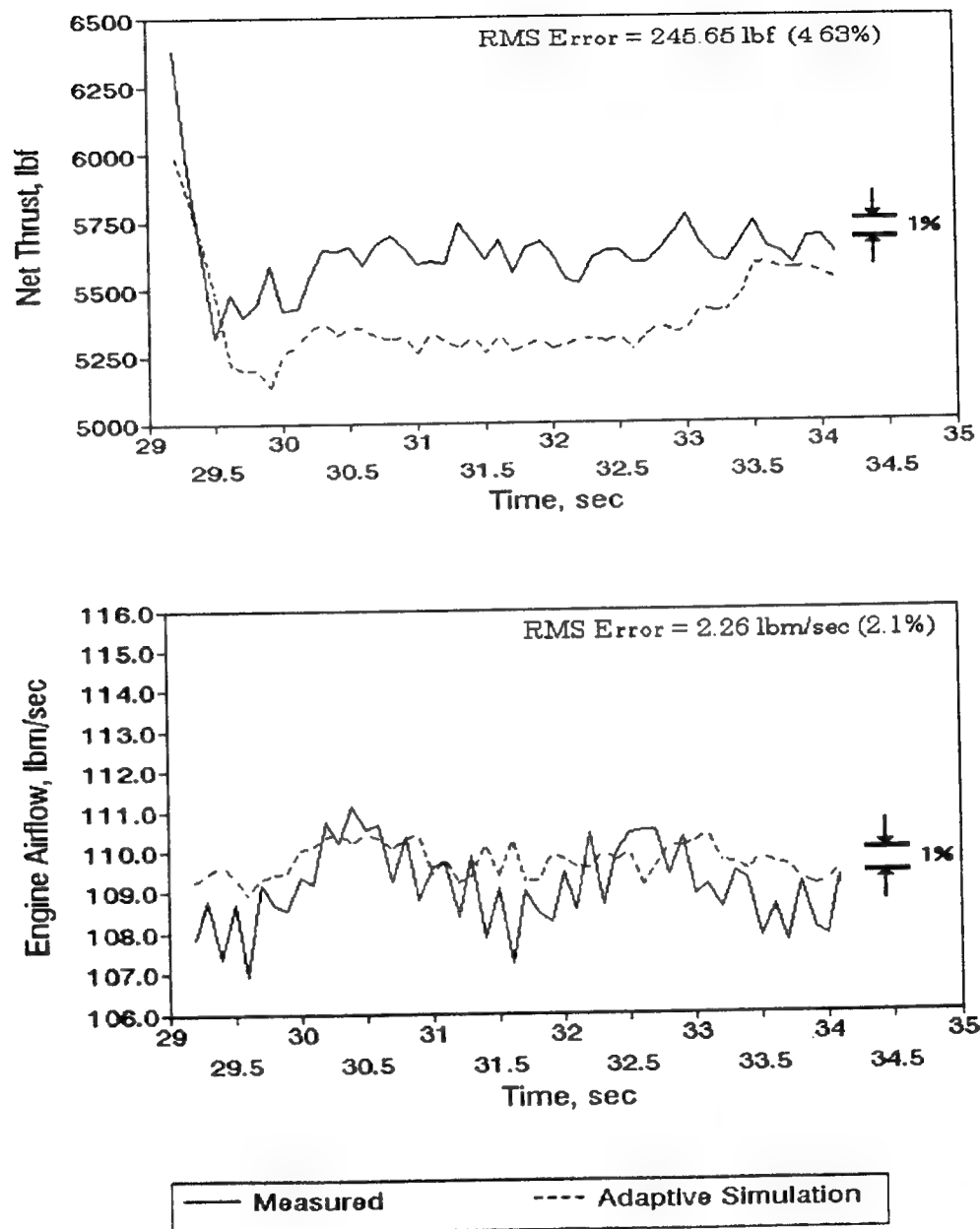

Engine Airflow

Figures A10 & A11. Engine test data.

Figures A12 & A13. Comparison of actual measurements versus ATEST
design point calculation.

Figures A14 & A15. Comparison of actual measurements versus SCIDNT estimated parameters with nominal weighting factors.

Figures A16 & A17. Comparison of actual measurements versus SCIDNT
estimated parameters with a modified airflow
weighting factor (W = .01)

DESIGN AND TEST OF A DIGITAL CROSSBAR SWITCH
FOR TMS320C40 DIGITAL SIGNAL PROCESSORS

Bryant E. Sorensen
Graduate Research Assistant
Department of Electrical and Computer Engineering

University of Tennessee Space Institute
B.H. Goethert Parkway
Tullahoma, TN 37388

DESIGN AND TEST OF A DIGITAL CROSSBAR SWITCH
FOR TMS320C40 DIGITAL SIGNAL PROCESSORS

Bryant E. Sorensen
Graduate Research Assistant
Department of Electrical and Computer Engineering
University of Tennessee Space Institute

ABSTRACT

A digital crossbar/crosspoint (Xbar) switch has been designed for the communication ports (commports) of the Texas Instruments TMS320C40 Digital Signal Processor (DSP). The Xbar switch allows any one of eight commports attached to the switch to communicate with any of the other seven commports, with a maximum of four connections in the switch at a time. The crosspoint switch allows any one port to broadcast data to as many of the other seven ports as desired. This Xbar design was reviewed, laid out on a printed circuit board, built, and the crossbar part of the switch tested during the 1993 AFOSR summer research tour. This report discusses the design of the switch and the software routines used to test and control the switch. These software routines form the basis for a complete Xbar switch control software package. The feasibility of using this switch in a TMS320C40 parallel processing system is demonstrated by these tests.

DESIGN AND TEST OF A DIGITAL CROSSBAR SWITCH
FOR TMS320C40 DIGITAL SIGNAL PROCESSORS

Bryant E. Sorensen

I. INTRODUCTION

The Texas Instruments TMS320C40 Digital Signal Processor (DSP) has six high-speed communication ports (commports) on the chip. This presents a great advantage in parallel processing systems, as each processor can transmit and receive data from up to six other processors quickly. However, in larger systems that handle data acquisition and analysis, it is advantageous to be able to communicate with more than six processors for data comparison (such as correlations) or processing speed-up. A *crossbar switch (Xbar)* has been designed which allows any one of eight DSP's connected to it to communicate with any one other DSP through a commport on each DSP, with a maximum of four connections in the switch at a time. A *crosspoint switch* has also been integrated in the system which provides the ability for one DSP to broadcast data from its commport to as many of the other seven ports as desired. This report reviews the design and details the testing of the crossbar part of the switch.

1. Discussion of Problem

Both the crossbar and crosspoint parts of the switch must be able to connect any two of the eight data busses in the system. Each connection must be completed in minimum time so the system does not become bottlenecked.

The crossbar switch must handle the handshaking to change the data bus direction. Each commport has a bus request line which it asserts when it wants

ownership of the bus in order to transmit data. Each port has a bus grant signal which it asserts when it has finished transmitting data and is ready to receive data. These signals must be handled so the switch can change the bus direction to be ready when the commports change their direction and avoid electrical damage to the DSP's.

The crosspoint switch does not allow direction changes, but instead must logically combine the multiple data handshaking signals (the 'data ready' lines) from the receiving ports to create one signal at the broadcasting port. This combination must only use those signals from the ports that are receiving data and ignore any ports which are not involved in the broadcast.

2. Methodology

The design was divided into five main parts:

(1) The data bus connection for both the crossbar and crosspoint switch;

(2) A bus direction handshaking arbiter for the crossbar switch;

(3) A data bus handshaking arbiter for the crosspoint switch;

(4) Address decoding for the different parts and system interconnects;

(5) A control unit for the entire device with software.

This report examines the design and measured performance of the Xbar switch, specifically the data bus connections, the bus direction handshaking arbiter, and the control unit with address decoding. Time did not allow for testing of the data bus handshaking arbiter and the crosspoint part of the switch. The results show the feasibility of using a crossbar/crosspoint switch in a C40 parallel processing system. Data transfer speeds through the switch were also measured.

II. THE TI TMS320C40 DIGITAL SIGNAL PROCESSOR'S COMMUNICATION PORTS (COMMPORTS)

1. Communication Port Signals and Their Operation

The Texas Instrument TMS320C40 (C40) is a 32-bit floating point digital signal processor (DSP). It has the unique feature of six communication ports (commports) on the chip. Each commport consists of twelve lines – eight data lines, a data strobe ($\overline{CSTRB}$), a data acknowledge ($\overline{CRDY}$), a bus request ($\overline{CREQ}$), and a bus acknowledge ($\overline{CACK}$). Each of these lines is bidirectional. The data, $\overline{CSTRB}$, and $\overline{CACK}$ lines are driven when the commport is in output mode, with $\overline{CRDY}$ and $\overline{CREQ}$ being driven in the input mode. The $\overline{CSTRB}$ and $\overline{CRDY}$ lines provide common handshaking for the data, with $\overline{CSTRB}$ being driven low by the output port when valid data is present, and $\overline{CRDY}$ being asserted low by the input port when the data has been received. The $\overline{CREQ}$ and $\overline{CACK}$ lines are used to arbitrate data bus ownership between two connected ports. The input port will assert the $\overline{CREQ}$ line low when it has a non-empty output buffer and its output has not been halted by the DSP's CPU. The output port will then drive $\overline{CACK}$ low when it has finished transferring a 32-bit word (four consecutive byte transmissions) or if its output buffer is empty. The input port then detects the $\overline{CACK}$ and removes the $\overline{CREQ}$ signal. This drives the $\overline{CACK}$ line high at the output port and signals the other lines to switch directions. The input port then becomes the output port and vice versa. (For further detail on commport operation the reader is directed to [5], chapter 8).

On power-up, half of the C40's commports are automatically configured as output ports (ports 0, 1, and 2) and half as input ports (ports 3, 4, and 5) ([5] pp. 8-14 and 8-15). No two input or two output ports should be directly connected as this can damage the C40.

III. COMMPORT LINE CONNECTIONS USING THE
LSI LOGIC CROSSBAR SWITCH

1. Hardware Features of the Crossbar Switch Part

The actual connections in the crossbar/crosspoint switch are implemented using the LSI Logic L64270 64x64 Crossbar Switch (Xbar) chip. This chip allows any one of 64 outputs to be connected to any one of 64 inputs, to be set to a constant value (high or low), or to be tri-stated. Each output has a 64-to-1 mux and registers which select the input line to which it will connect. All outputs can be disabled at once also. The chip supports a bidirectional mode, in which each input is hardwired to a corresponding output. The Xbar allows for groups of ports to be treated as a single entity. These groups must be a power of two in size, ranging from width 1 to width 64. This design uses the Xbar in its bidirectional mode.

To avoid confusion, each input and output on the Xbar will be referred to as a 'line' while the combination of a pair in the bidirectional mode will be called a 'port'. A grouping of these ports will be called a 'bus'.

The Xbar chip has 19 control lines. The lines CI.0-CI.5 are used in initialization to indicate the bus width to be used. After initialization, they are used to select the input lines in the 64-to-1 mux. CI.6 is asserted in initialization to indicate the bidirectional mode, and afterwards is kept low unless the output is to be set at a constant or disabled. CI.7 is not used in initialization and indicates pipeline mode otherwise. It is tied to ground since pipelined mode will not be used in this design. The lines REGADR.0-REGADR.5 select the output bus that is being configured. REGADR.6 is asserted to indicate initialization of the Xbar. The collections CI.0-CI.6 and REGADR.0-REGADR.6 will be referred to as CI and REGADR respectively. BDCYC1 is used in the bidirectional mode to indicate the configuration cycles (explained below). $\overline{WE}$ is the control line data strobe.

BNKLDI is a latch signal used to load the configuration data from a holding memory to the individual output control registers. $\overline{OE}$ is the global output enable signal ([2] p. 187).

2. Configuration of the Xbar chip

To initialize the Xbar chip, REGADR.6 is asserted and the bus width to be used is specified on CI.0-CI.5 ([2] p. 189). CI.6 is asserted to indicate bidirectional mode is desired. REGADR.0-REGADR.5 and CI.7 are don't cares during the initialization. $\overline{WE}$ is then pulled low for at least 10 nsec. (For more complete timing of the initialization and configuration timing, see [2] p. 192).

Connections in the bidirectional mode take two cycles to complete. During the first cycle. BDCYC1 is held high while specifying the output port or bus address on REGADR and the input port on CI. $\overline{WE}$ is pulled low, then returned high. This first cycle disables the output line of the input port so as to not damage the output line driver. In the next cycle. BDCYC1 is taken low, REGADR and CI remain the same, and $\overline{WE}$ is pulsed low again. This cycle makes the desired connection between the input line of the input port and the output line of the output port. If desired. BNKLDI can be held low during both cycles. then pulled high any time after completion to latch the configuration. This allows for multiple connections specified at the Xbar to be latched in at the same time; for instance, if all four connections were to be made in the Xbar before latching them in to start data transfers. ([2] pp. 188-189).

The Xbar chips were placed in parallel with respect to the control signals to speed up configuration of the switch. However, since the $\overline{CRDY}$ and $\overline{CREQ}$ lines run opposite to the other ten lines from the commport. the Xbar chip used to make their connections switched the CI and REGADR from the other Xbar chips used to connect the Data, $\overline{CACK}$. and $\overline{CSTRB}$ lines.

8-7

IV. BUS ARBITRATION HANDLING OF $\overline{CREQ}$ AND $\overline{CACK}$

The C40 commports use the $\overline{CREQ}$ and $\overline{CACK}$ lines to arbitrate ownership of each data bus connection. As explained before, each port starts as either an output port with bus ownership or as an input port without bus ownership at power-up. However, it may be desired to connect two ports which conflict in bus ownership. This can create electrical problems ([5] pp. 8-14,8-15) as well as data loss. The $\overline{CREQ}/\overline{CACK}$ arbitration unit provides both electrical isolation to prevent harm to the C40's as well as a go-between to assure correct bus direction and ownership.

The arbiter seeks to assure that the $\overline{CREQ}$ will be responded to quickly enough and that turnaround occurs quickly enough so that the commports, the Xbars, and the other chips are not damaged in the process. It also works to make changes of direction as quickly as possible to achieve maximum data throughput. This is all handled in hardware so the control unit does not need to be involved, which would slow down the process. The arbiter also handles reconfiguration requests from the control unit by assuring that data has stopped flowing before the OK is given to reconfigure the system. This should prevent data from being lost.

1. Bus Transfer Sequence and Timing

The bus transfer sequence is as follows. The input port of a connected pair of commports asserts $\overline{CREQ}$ when it has data to send. When the output port has completed its data transfer or is not sending data, it asserts $\overline{CACK}$. The input port detects $\overline{CACK}$ and removes $\overline{CREQ}$. This drives $\overline{CACK}$ back high, at which time the commport signals change direction.

The timing for the transfer of data bus ownership (called "token transfer") is given in [5], pp. 14-33 to 14-36. Notice, however, that these are relative timings to the internal synchronizers of the C40's. As far as this author knows, no solid absolute timing is known for the bus token transfers. The timings given do indicate

some idea of what two commports would do if directly connected and this design tries to mimic these as nearly as possible. The times given will have to be stretched in many cases to allow the control unit to respond, however.

2. State Machine Design for Bus Direction Switching

A state machine for $\overline{CREQ}$ and $\overline{CACK}$ handling was written in Altera Hardware Description Language (AHDL) (see [4]). The corresponding state machine diagram is included in Figure 1. The OUT signal to the state machine is created by decoding the address lines of the output port being connected. This address comes from the three upper bits of CI at the Xbar chips. Similarly, the IN signal comes from the three upper bits of REGADR. The 'O' and 'I' inputs are data lines which give the output and input ports' status, i.e., if the desired output port is already in the output state, O is asserted and likewise for 'I'. These status bits are only used on the initial configuration sequence since the arbitration state machines keep track of each port's state afterwards. CONFR denotes 'configuration request' and is asserted to request a reconnection during operation (not at initial configuration of the switch). BC is the 'broadcast' signal, which is asserted when $\overline{CREQ}$ and $\overline{CACK}$ are to be ignored after the connection is made (crosspoint switch). LTCH is a latching signal which is the same as BNKLDI at the Xbars. It is handshaking from the control unit used to show that a connection has been completed.

$\overline{CREQ}$ and $\overline{CACK}$ are the commport signals into the arbitration unit, while $\overline{CRX}$ and $\overline{CAX}$ are the request and acknowledge signals fed to the Xbars, respectively. The $\overline{CREQ}$, $\overline{CACK}$, $\overline{CRX}$, and $\overline{CAX}$ signals all go through tri-state buffers from this chip so that they may be turned on and off. They are all pulled up through resistors to provide a default high state when disabled. These four outputs and their enabling signals are asserted as shown in Table 1. By separating these signals and tri-stating them when necessary, the state machine can respond more

quickly to $\overline{CREQ}$ and $\overline{CACK}$ and provide electrical isolation.
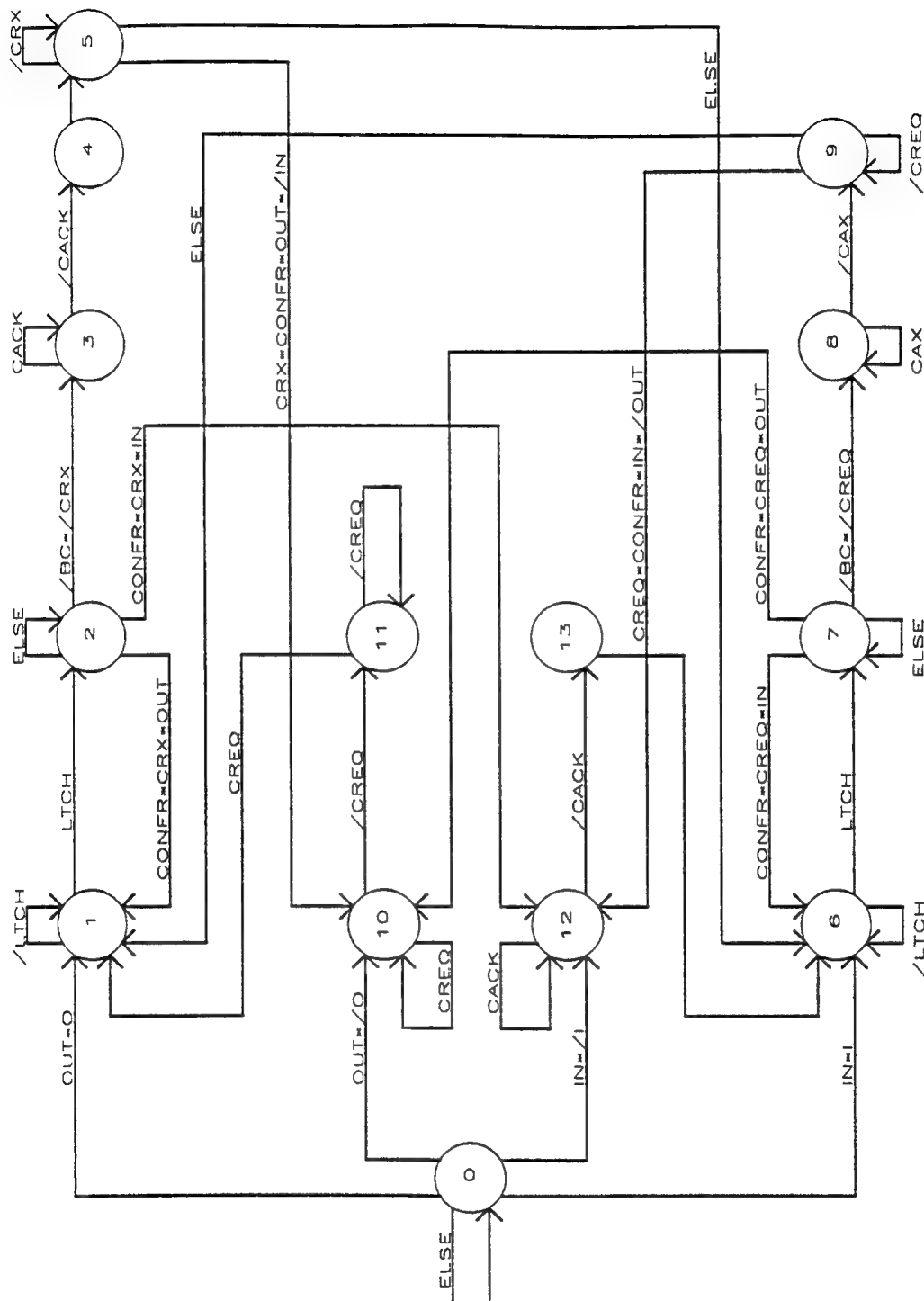

FIGURE 1 – STATE MACHINE DIAGRAM FOR $\overline{CREQ}/\overline{CACK}$ ARBITRATION

Table 1 – Output Table for $\overline{CREQ}/\overline{CACK}$ State Machine

STATE	CREQ	CREQ ENA	CACK	CACK ENA	CRX	CRX ENA	CAX	CAX ENA
s0	1	0	1	0	1	0	1	0
s1	1	0	1	0	1	0	1	0
s2	1	1	1	0	1	0	1	1
s3	0	1	1	0	1	0	1	1
s4	0	1	1	0	1	0	0	1
s5	1	1	1	0	1	0	0	1
s6	1	0	1	0	1	0	1	0
s7	1	0	1	1	1	1	1	0
s8	1	0	1	1	0	1	1	0
s9	1	0	0	1	1	1	1	0
s10	1	0	1	1	1	0	1	0
s11	1	0	0	1	1	0	1	0
s12	0	1	1	0	1	0	1	0
s13	1	1	1	0	1	0	1	0

OK is a handshaking signal to the control unit indicating that the commports are ready to be connected as far as the $\overline{CREQ}$ and $\overline{CACK}$ signals are concerned. The OK signal is only asserted in State 1 and State 6, when the state machine is looking for the LTCH signal indicating that the connection is made by the control unit. The control unit will be able to read these OK lines to determine when the data has stopped flowing and when the ports are in the desired states.

Each port has its own state machine to handle its $\overline{CREQ}$ and $\overline{CACK}$ signals. All eight state machines are then connected with proper address decoding going to the OUT and IN state machine lines.

4. Software Control of the Arbitration Unit

A connection on this chip is configured at the exact same time as the Xbars. This is made possible by the overlap in addresses at the decoder (see Sec. V).

The exact same addresses are used and the same $\overline{WE}$, BDCYCL, and BNKLDI signals are used. The $\overline{WE}$ at the arbitration chip enables the input and output port addresses at the D flip-flops so that the chip may have time to make necessary changes before allowing the connection to be completed. The control unit may make multiple connections without receiving the necessary OK signals to complete them, but none of these connection will be latched into the Xbar and this arbitration chip by BNKLDI going high until all the OK signals the controller is looking for have been received. The control unit will assert a 'read OK' (RDOK) signal, and then will have access to the OK's from each port on the upper byte of its data bus. The BRDCST signal may also be asserted if broadcast mode is desired.

When a reconfiguration is desired, the CONFR line is asserted with the addresses of the two ports involved being latched in as well. The broadcast status is also latched in at the BC line. The control unit then asserts RDOK. When the ports have stopped transferring data and are going the correct direction (i.e., output or input state), the two ports specified will have their OK signals asserted. The control unit then reads all the OK lines, checks for the two from the ports to be reconfigured, then makes the changes at the Xbars when both OK's are received. This assures that no data will be lost and that improper changes are not made.

V. ADDRESS DECODING FOR CROSSBAR SWITCH

Address lines from the control unit are used to address the Xbar chips, the $\overline{CRDY}$ arbiter, the $\overline{CREQ}/\overline{CACK}$ handler, the $\overline{CREQ}$ buffer, and provide other control signals to the $\overline{CRDY}$ arbiter, $\overline{CREQ}/\overline{CACK}$ handler, and the Xbar chips. The design uses address lines instead of data lines to provide these control signals because the address lines are always in a determined state and not tri-stated as the data lines are between data transfers.

Nine address lines (A0-A8) from the control unit are used both to create control

signals and for decoding to access the different parts of the switch. A0 is used to create the BDCYCL signal to the Xbars and A1 becomes BNKLDI. A2 is used for CONFR and A3 for BRDCST for the $\overline{CREQ}/\overline{CACK}$ arbitration unit. A4, A5, and A6 are used for ADR0, ADR1, and ADR2, respectively, by the $\overline{CRDY}$ arbiter (this report does not cover the $\overline{CRDY}$ arbiter). A7 and A8 are a0 and a1, respectively, to the address decoder explained below. The address decoder also uses A2 (CONFR). The decoder uses the $\overline{IOE}$ signal as the data strobe, and the $\overline{W}$ line as R/W (read asserted high, write asserted low) signal.

The address decoder needs to provide 6 signals. They are $\overline{RDCRQ}$ (read the $\overline{CREQ}$ lines), $\overline{WER}$ (write to $\overline{CRDY}$ arbiter for broadcast setup), $\overline{WEQ}$ (write to $\overline{CREQ}/\overline{CACK}$ handler), RDOK (read the OK signals from the $\overline{CREQ}/\overline{CACK}$ handler), $\overline{OEX}$ (enable/disable the $\overline{CRDY}$ Xbar chip), and $\overline{WEX}$ (write to the Xbar chips). The decoding for these signals and the memory map of the Xbar switch are shown in the figures below.

```
PIN   2     CONFR    INPUT           ; configure request
PIN   5     A0       INPUT           ; low address line
PIN   7     RW       INPUT           ; Read/Write line
PIN   9     STRB     INPUT           ; Data strobe
PIN  11     A1       INPUT           ; high address line
PIN  15     SET      CMBFBK          ; set for OEX
PIN  16     RST      CMBFBK          ; reset for OEX
PIN  17     WEX      COMBINATORIAL   ; write to Xbars
PIN  18     OEX      CMBFBK          ; enable/disable Xbar for /CRDY,/CREQ
PIN  19     RDOK     COMBINATORIAL   ; read the OK lines at /CREQ,/CACK chip
PIN  20     WEQ      COMBINATORIAL   ; write to /CREQ,/CACK chip
PIN  21     WER      COMBINATORIAL   ; write to /CRDY chip
PIN  22     RDCRQ    COMBINATORIAL   ; read the /CREQ lines

; logic equations

EQUATIONS

/WEX = /RW * /STRB * /A1 * /CONFR
/WER = /RW * /STRB * /A0 * A1
/WEQ = /RW * /STRB * A0 * /A1
     + RW * /STRB * /A0 * /A1
RDOK = RW * /STRB * /A0 * A1
SET = /RW * /STRB * A0 * A1
RST = RW * /STRB * A0 *A1
OEX = /SET * RST + /SET * OEX
/RDCRQ = RW * /STRB * A0 * /A1
```

FIGURE 2 – ADDRESS DECODING EQUATIONS FOR XBAR SWITCH

(C)

EXPANDED MAP FOR WRITE
TO CREQ/CACK AND
XBARS FOR CONFIGURATION

Addr	
0FF / 090	7 Reflections of block below
08F	brdcst * confr * bnkldi * bdcyc1
08E	brdcst * confr * bnkldi * bdcyc1
08D	brdcst * confr * bnkldi * bdcyc1
08C	brdcst * confr * bnkldi * bdcyc1
08B	brdcst * confr * bnkldi * bdcyc1
08A	brdcst * confr * bnkldi * bdcyc1
089	brdcst * confr * bnkldi * bdcyc1
088	brdcst * confr * bnkldi * bdcyc1
087	brdcst * confr * bnkldi * bdcyc1
086	brdcst * confr * bnkldi * bdcyc1
085	brdcst * confr * bnkldi * bdcyc1
084	brdcst * confr * bnkldi * bdcyc1
083	brdcst * confr * bnkldi * bdcyc1
082	brdcst * confr * bnkldi * bdcyc1
081	brdcst * confr * bnkldi * bdcyc1
080	brdcst * confr * bnkldi * bdcyc1

(B)

EXPANDED MAP FOR WRITE
TO CRDY ARBITER

Addr	
17F / 178	Port 7 broadcasts
170	NO BROADCAST
168	Port 6 broadcasts
160	NO BROADCAST
158	Port 5 broadcasts
150	NO BROADCAST
148	Port 4 broadcasts
140	NO BROADCAST
138	Port 3 broadcasts
130	NO BROADCAST
128	Port 2 broadcasts
120	NO BROADCAST
118	Port 1 broadcasts
110	NO BROADCAST
108	Port 1 broadcasts
100	NO BROADCAST

(A)

MAIN MEMORY MAP

(offsets from 9000 8000h, start of DSP-Link)

Addr	
7FFF / 200	Not Used (reflections of spaces below)
1FF	Write to disable CRDY xbar
180	Read to enable (OEX)
17F	Write to CRDY arbiter (WER)
100	Read OK's from CREQ / CACK chip (RDOK) (See (B))
0FF	Write to CREQ/CACK chip and to Xbars (WEQ, WEX)
080	Read CREQ buffer (RDCRQ) (See (C))
07F	Write to Xbars for initialization (WEX)
000	Read to latch connections (WEQ)

FIGURE 3 – MEMORY MAP OF XBAR SWITCH IN CONTROL UNIT

8-14

VI. SOFTWARE ROUTINES FOR CONTROL

The crossbar switch was laid out on a VME-size board, ordered, and put together this summer. The design of the Xbar switch was also reviewed, although the main part of the design took place before this AFOSR research tour. The board was then tested with the five test software routines given in the following pages. These algorithms were tested using a Spectrum Signal Processing PC-host C40 board with a DSP~Link interface (for detailed information on DSP~Link, the reader is referred to [3]) and the accompanying debugger. DSP~Link is a hardware interface protocol which can expand memory on the Spectrum board and/or provide communication between other DSP elements. DSP~Link resides in the C40's global memory starting at address 9000 8000 Hex, as shown in Figure 4 of the preceding section, and provides 16 data lines.

These routines show the basic commands necessary to configure the Xbar switch as well as check the data throughput and integrity through the Xbar. These routines are merely rough sketches of the necessary parts of control code for the Xbar and need to be integrated into a more complete software package to be useful in a parallel processing systems. Each routine has comments about its use and the part of the Xbar it tests.

The meaning of each configuration word is given in the template below.

CREG/CACK chip		I2	I1	I0				IS		O2	O1	O0				OS
Xbars	RG6	RG5	RG4	RG3	RG2	RG1	RG0		CL6	CL5	CL4	CL3	CL2	CL1	CL0	
CRDY chip										U7	U6	U5	U4	U3	U2	U1	U0
Reads	OK7	OK6	OK5	OK4	OK3	OK2	OK1	OK0	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

RGx = REGADRx
IS = ISTAT (Input Port Status)
OS = OSTAT (Output Port Status)
Ux = Data Lines - Indicates Port x will receive broadcast
OKx = OK Lines to Read from CREG/CACK chip
CRx = CREGx for Reads from CREG Buffer Chip

FIGURE 4 – CONFIGURATION WORDS TEMPLATE

```
;  PROG1.ASM
;  Check the connection of two commports through the Xbar
        .data
XBARI    .word    90008000h            ; address to initialize xbar chips
LATCH    .word    90008002h   ; address to latch in connections
CRDYEN   .word    90008180h   ; address for enabling /CRDY xbar
RDOK     .word    90008100h   ; address to read the OK lines
CYCLE1   .word    90008081h   ; 1st cycle of connection write, non-broadcast
BASECD   .word    002ff800h   ; base address for start of code
GLOBINT  .word    32778010h   ; global memory control word
CONNECT  .word    00006121h   ; connection to be made
comm_port0_ctl .word 00100040h; Port 0 control
comm_port0_inp .word 00100041h;    input
comm_port0_otp .word 00100042h;    output
comm_port3_ctl .word 00100070h; Port 3 control
comm_port3_inp .word 00100071h;    input
comm_port3_otp .word 00100072h;    output
    .text
   LDP    @BASECD,DP            ; set stack and data pointers
   LDA    @BASECD,SP
   LDHI   0010h,AR0             ; load address for global memory interface reg
   LDI    @GLOBINT,R0           ; load global memory interface control word
   STI    R0,*AR0               ; set the global mem interf reg
; create the OK mask
   LDI    @CONNECT,R3           ; set up connect address
   LDI    -4,R4                 ; shift counter, get 2nd nibble
   LDI    -12,R5                ; shift counter, get 4th nibble
   LDI    0100h,AR1            ; number to be shifted
   LDI    070h,R6              ; mask for 2nd nibble
   LDI    07000h,R7            ; mask for 4th nibble
   AND    R3,R6,R0             ; get 2nd nibble
   AND    R3,R7,R1             ; get 4th nibble
   LSH    R4,R0                ; shift down to right 4 bits
   LSH    R5,R1                ; shift down right 12 bits
   LSH    R0,AR1,R0            ; shift 1 left by R0 bits (max 7)
   LSH    R1,AR1,R1            ; shift 1 left by R1 bits (max 7)
   OR     R0,R1,R6             ; OR results and store in R6
; load control registers for Xbar switch
   LDA    @XBARI,AR0            ; Xbar initialization address
   LDA    @RDOK,AR2             ; address to read OK's
   LDA    @CRDYEN,AR3           ; address to enable/disable /CRDY xbar
   LDA    @LATCH,AR4            ; latch signal for connections
   LDA    @CYCLE1,AR5           ; Xbar address for 1st cycle of configure write
; initialize the Xbar chips
   STI    R0,*AR3               ; dummy write to enable /CRDY Xbar
   LDI    8E8Eh,R0              ; control word indicating 8-wide ports at Xbar
   STI    R0,*AR0               ; initialize the Xbars
; make one-one connections
   STI    R3,*AR5--(1)          ; make the first connection cycle
   STI    R3,*AR5++(IR1)        ; second connection cycle keeping BNKLDI low
testok:
   AND    R6,*AR2,R1            ; read OK lines (AR2->OK address,R6=OK mask)
   SUBI   R6,R1                 ; check if OK's are asserted
   BNZ    testok                ; keep checking if not both asserted
   LDI    *AR4,R4               ; if both OK's asserted, latch in connection
                                ;   by changing BNKLDI to high (dummy read)
   LDA    @comm_port0_ctl,AR0  ; Load comm port 0 cont reg addr
   LDA    @comm_port0_otp,AR1  ;        output FIFO addr
   LDA    @comm_port3_ctl,AR2  ; Load comm port 3 control reg. address
   LDA    @comm_port3_inp,AR3  ;        input FIFO address
   LDHI   5555h,R1              ; Output word setup
   OR     5555h,R1
   AND    0DFFFh,ST             ; CPU global interrupt disabled
   LDI    0100h,R0              ; Mask for bit 8 of CPCR, port 0
   LDI    1E00h,R2              ; Mask for input level of port 3
OUT: TSTB  *AR0, R0             ; Check if O/P FIFO is full
   BNZ    OUT                   ; If yes, check again
   STI    R1, *AR1              ; write value to O/P, port 0
IN: AND    *AR2,R2,R3           ; check input level of port 3
   BZ     IN                    ; keep checking if input FIFO is empty
   LDI    *AR3,R3               ; load value from I/P, port 3
   SUBI   R1,R3,R4              ; check that the output and input values are =
   BZ     OUT                   ; if they are, keep outputting
STOP: BR    STOP                ; otherwise stop
```

8-16

```
;  PROG2.ASM
;  Test of commport data speed by polling method
COMM_PORT0_CTL .word 00100040h   ; Port 0 control
COMM_PORT0_DAT .word 00100042h   ; port 0 output FIFO
COMM_PORT3_DAT .word 00100071h   ; Port 3 input FIFO
LOOP_COUNT     .word 12500000    ; number of words to send (4 bytes each)
START:  LDP   COMM_PORT0_CTL
        LDA   @COMM_PORT0_CTL,AR1
        LDA   @COMM_PORT0_DAT,AR2
        LDA   @COMM_PORT3_DAT,AR3
        LDA   @LOOP_COUNT,AR4
        AND3  0EFh,*AR1,R9   ; set up port 0 for output
        STI   R9,*AR1
        LDHI  55AAh,R0     ; word to be output
        OR    55AAh,R0
        STI   R0,*AR2      ; output 8 words to fill port 0's output FIFO
        STI   R0,*AR2
        STI   R0,*AR2
        STI   R0,*AR2
        STI   R0,*AR2
        STI   R0,*AR2
        STI   R0,*AR2
        STI   R0,*AR2
WAIT1   LSH   -8,*AR1,R1 ; wait for FIFO to be 1/2 full
LOOP    BC    WAIT1
        STI   R0,*AR2      ; store 4 more words
        STI   R0,*AR2
        STI   R0,*AR2
        STI   R0,*AR2
        LDI   *AR3,R1      ; read the transferred words from input FIFO
        LDI   *AR3,R1      ;    of port 3
        DBD   AR4,LOOP     ; decrement the count
        LDI   *AR3,R1
        LDI   *AR3,R1
        LSH   -8,*AR1,R1  ; check output FIFO level again
STOP    BR    STOP         ; endless loop when done (breakpt)


;  PROG3.ASM
;  Check state machines ability to change commport connection direction
RECONN   .word    00002060h   ; switch direction of ports input->output, vv
*** SAME AS PROG1 UP TO HERE ***
    LDI   @RECONN,R7         ; load connection change, OK mask is same
    LDA   @comm_port0_inp,AR1 ; changing over from output to input
    LDA   @comm_port3_otp,AR3 ; changing input to output
    LDI   1,R3               ; token showing transfer successful
; make change in direction by outputting to port 3
OUT3: TSTB  *AR2,R0          ; check if o/p buffer full
   BNZ   OUT3                ; if so, check again
   STI   R3,*AR3             ; write out to port 3, initiate change
   STI   R7,*AR5--(1)        ; make the first connection cycle
   STI   R7,*AR5++           ; second connection cycle keeping BNKLDI low
testok: AND  *AR6,R6,R1      ; read OK lines (AR6->OK address,R6=OK mask)
   SUBI  R6,R1               ; check if OK's are asserted
   BNZ   testok              ; keep checking if not both asserted
   LDI   *AR4,R4             ; latch in connection
IN0: AND   *AR0,R2,R1        ; check input level of port 0
   BZ    IN0                 ; keep checking if input FIFO is empty
   LDI   *AR1,R1             ; load value from I/P, port 0
   SUBI  1,R1                ; check for correct value
   BNZ   ERROR               ; go to error if not = 1
STOP: BR    STOP             ; otherwise stop
ERROR: BR   ERROR            ; reaches here on an error


;  PROG4.ASM
;  Check for Reconfigure ability of state machines
CONFR   .word    90008085h         ; reconfigure request
RECONN  .word    00002031h         ; reconnection
comm_port1_ctl .word 00100050h     ; commport 1 control
comm_port1_otp .word 00100052h     ; port1 output FIFO
; create the OK mask for the 2nd connection
    LDI   @RECONN,R7               ; set up connect address
    ...       (same procedure as in PROG1)
    OR    R0,R3,R6                 ; OR results and store in R6
*** Make connection, check the data flow as in PROG1 ***
    LDI   @CONFR,AR7               ; address to request reconfigure
    LDA   @comm_port0_inp,AR1   ; changing over from output to input
    LDA   @comm_port1_ctl,AR2   ; goto port 1
    LDA   @comm_port1_otp,AR3
```

8-17

```
; make the reconfigure request
    STI    R7,*AR7              ; make the request -- port 1 while be init'ed
testok: AND    *AR6,R6,R1       ; check for the OK to come back
    SUBI   R6,R1                ; see if it matches the mask
    BNZ    testok
    STI    R7,*AR5++(1)         ; make the connection
    STI    R7,*AR5--(1)
    LDI    *AR4,R4              ; dummy read to latch in connection
    LDI    *AR6,R5              ; dummy read (check OK's) to get latch off
*** Check new connection as in PROG1 ***


;  PROG5.ASM
;  Check state machine's ability to turn ports around at initial connection
*** SAME AS PROG1 EXCEPT FOR CONNECTION WORD SHOWN BELOW ***
    .data
CONNECT  .word    00002060h    ; NOTE THAT BOTH PORTS HAVE TO BE TURNED AROUND
```

VII. RESULTS

The board was first tested the ensure that the parallelization of the crossbar
chips would work. This proved to be the case. The three chips can all be pro-
grammed for a certain connection in two or three cycles (two if the connection
flows through instead of being latched in after it is written). These cycles must
be accompanied by the handshaking necessary to ensure the state machines are
responding, specifically by checking the OK lines from the $\overline{CREQ}/\overline{CACK}$ chip.
With this test, it was also verified that the data, $\overline{CACK}$, and $\overline{CSTRB}$ lines
can flow opposite the $\overline{CREQ}$ and $\overline{CRDY}$ lines, thus proving the concept of the
parallelization of the LSI Logic Crossbar chips, at least in the bidirectional mode.

Next, the data flow rate was checked using an efficient polling method with
the commports. The polling only took place every four words and is comparable
to DMA for speed. In multiple tests, the crossbar achieved data throughput on the
range of 7.2 MBytes/sec to 7.5 Mbytes/sec. This compares to the approximately
12 MBytes/sec that similar tests provide on straight commport-to-commport con-
nections with the approximately 1 meter of cable used. The dropoff in speed comes
from the propagation delay of the signals through the Xbar chips, which can reach
25 nsec each way. These data rates are encouraging since they do not even cut in
half the original speed of the link in this test.

The state machines for $\overline{CREQ}/\overline{CACK}$ handling were then further tested, specifically to check for direction changing between two connected commports. Again, the results were favorable. Although the TI data book shows relative timing for these bus ownership transfers, it was discovered that much more leeway exists for the $\overline{CREQ}$ and $\overline{CACK}$ signals that one might expect from studying the data book. This is very encouraging as it indicates the ability of slower PLD's to keep up with the C40. The timing for the $\overline{CRX}$ and $\overline{CAX}$ signals which come from the $\overline{CREQ}/\overline{CACK}$ chip and go to the Xbars can be improved, and hopefully with the next design revision this will be corrected.

The next test demonstrated favorably that the Xbar can reconfigure during operation (on-the-fly). Two ports were connected, then one was requested to reconnect to another port after turning around. The data was once again able to go across the Xbar without corruption after both the original and the second connections were made.

Finally, the state machine demonstrated that it can turn around ports at initialization. It was requested to change both an input port to an output, and an output port to an input. The connection was made successfully and data was again sent uncorrupted through the crossbar.

VIII. CONCLUSIONS AND FUTURE WORK

The crossbar switch has been proven in a limited manner to perform satisfactorily in allowing arbitrary connections between any two commports attached to the switch. Data transfers occur uncorrupted at a maximum tested speed of 7.5 MBytes/sec for polled commport transfers, as compared to 12 Mbytes/sec for direct commport connections under the same algorithm. The switch should become a useful tool in parallel processing systems using the TI TMS320C40, allowing topologies to be changed in the system during processing.

8-19

Time did not allow for testing of the crosspoint switch part of the design, but this is a much simpler part of the switch; this part will be tested soon in the future.

Much remains to be done in developing the Xbar for further use. The software algorithms need to be integrated into a program which can run independently so the controlling C40 can automatically monitor the switch and make the changes. The software also needs to be optimized for speed to make the switch even more useful. A host PC interface is being developed now so that the switch can be controlled in a user-friendly environment. The switch can also be expanded to 16, 32, or 64 ports. Work has already been done in the design of these larger switches, but the concepts proved in this smaller 8-port version carry over quite nicely to the larger switches. The actual hardware implementation, debugging and checking of larger switches remains for the future.

BIBLIOGRAPHY

[1] "Data Book", Altera Corporation, Sep. 1991.

[2] "Digital Signal Processing (DSP) Databook", LSI Logic Corporation, Sep. 1991. pp. 186-199.

[3] "DSP~Link Technical Application Note", Issue 1.32, Spectrum Signal Processing Inc.. Burnaby, B.C.. Canada, Nov. 1991.

[4] "MAX+plusII Text Editor & AHDL", Altera Corporation, version 1.0, April 1991.

[5] "TMS320C4x User's Guide", Texas Instruments Corporation, May 1991.

INCOMPRESSIBLE NAVIER-STOKES FLOW ANALYSIS
FOR THE ARNOLD ENGINEERING DEVELOPMENT CENTER
HEAT-H1 TEST UNIT NOZZLE COOLING PASSAGE

Michael A. Weaver, MSAE
Graduate Research Assistant
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0150

Final Report for:
AFOSR Summer Research Program
Arnold Engineering Development Center

10 September 1993

INCOMPRESSIBLE NAVIER-STOKES FLOW ANALYSIS FOR THE ARNOLD ENGINEERING DEVELOPMENT CENTER HEAT-H1 TEST UNIT NOZZLE COOLING PASSAGE

Michael A. Weaver, MSAE
Graduate Research Assistant
School of Aerospace Engineering
Georgia Institute of Technology

Abstract

An incompressible Navier-Stokes flow solution was obtained for the Arnold Engineering Development Center HEAT-H1 Test Unit nozzle cooling passage. This flow solution is one component of a conjugate heat transfer algorithm developed by the author to predict heat load failure in the HEAT-H1 Test Unit. The HEAT-H1 Test Unit is a high enthalpy, arc-heated, free jet test facility. A backside water-cooled nozzle accelerates high temperature air to supersonic parallel flow at the exit. The research flow solver INS3D-UP (from NASA Ames Research Center) was used to solve the steady, incompressible Navier-Stokes equations for water flow on the backside of the nozzle. INS3D-UP uses an upwinded discretization scheme with the pseudo-compressibility solution method. An axisymmetric solution was obtained by solving the full three-dimensional system over three azimuthally rotated planes, while enforcing symmetry boundary conditions on the two outer planes. The results indicate the presence of recirculating flow and a stagnation point in a critical region of the cooled nozzle wall. Figures show velocity vectors and the pressure distribution for this critical region. The flow solution does not consider two-phase, subcooled boiling effects, which are expected to be present.

INCOMPRESSIBLE NAVIER-STOKES FLOW ANALYSIS
FOR THE ARNOLD ENGINEERING DEVELOPMENT CENTER
HEAT-H1 TEST UNIT NOZZLE COOLING PASSAGE

Michael A. Weaver

Introduction

The Arnold Engineering Development Center (AEDC) HEAT-H1 Test Unit is an arc-heated, free jet test facility, providing extremely high enthalpy air flow. Flows with enthalpies ranging from 2,000 to 8,500 Btu/lb_m and pressures ranging from 20 to 115 atm are routinely produced [1]. Mach numbers ranging from 1.8 to 3.5 are achieved with interchangeable nozzles. These nozzles are actively cooled with backside coflowing water undergoing subcooled nucleate boiling.

Proposed future applications for the HEAT-H1 require operating conditions beyond the current survival envelope. These applications need pressures of up to 200 atm, while nozzle wall heat load failures have been observed at pressures in the range 120 to 130 atm. The ability to identify and predict the heat load failure modes of the nozzle wall is crucial to extending operation of the HEAT-H1 beyond its current capabilities.

Heat load in the HEAT-H1 nozzle wall can be categorized as a coupled or conjugate heat transfer phenomenon. The conjugate heat transfer domain comprises coflowing high temperature air and subcooled water coolant on opposite sides of a copper-zirconium, converging/diverging nozzle. Heat transfer through the nozzle wall is characterized by solid body conduction with convection boundary conditions along the air side and water side of the nozzle wall. The air side heat transfer is characterized by forced convection with a turbulent boundary layer. The water side heat transfer is characterized by forced convection, subcooled, nucleate boiling. Convective heat transfer coefficients on each side of the nozzle wall are functions of the wall temperature and the respective flow properties, thus coupling the three regions of the domain.

The AEDC operating contractor for the HEAT-H1 (Calspan Corporation/AEDC Operations) has conducted a heat load failure analysis by assuming one-dimensional conjugate heat transfer through the nozzle wall [2, 3]. This approach gives good results for the thin-walled region of the nozzle, but failure analysis for the complete nozzle must consider multidimensional conjugate heat transfer. With sponsorship

from the Air Force Office of Scientific Research (AFOSR), the author pursued such a multidimensional approach while participating in the 1992 AFOSR Summer Research Program [4]. This multidimensional heat load failure analysis subsequently received additional funding through the 1993 AFOSR Summer Research Extension Program, and the 1993 AFOSR Summer Research Program.

The multidimensional approach requires numerical models (based on either analytic methods or engineering mathematical correlations) for the unsteady solid body heat transfer in the copper-zirconium nozzle, the flow properties and heat transfer coefficients on the air side of the nozzle, and the flow properties and heat transfer coefficients on the water side of the nozzle. During the 1992 AFOSR Summer Research Program, the author assembled the numerical models, developed a fully-coupled solution algorithm, and demonstrated the feasibility of the solution scheme. The 1993 AFOSR Summer Research Extension Program provided nine months of funding for refining the numerical models and solution algorithm.

The 1993 AFOSR Summer Research Program provides an additional three months of funding for improvement of the flow properties model on the water side of the nozzle. This effort seeks to obtain a complete incompressible, Navier-Stokes flow solution for the water side of the HEAT-H1 nozzle.

Discussion of Problem

In the HEAT-H1 Test Unit, tangentially injected air flows between an anode and a cathode for heating by an electric arc. The tangentially injected air creates swirling flow which serves to keep the electric arc from making contact with the wall in the region between the anode and cathode. The high temperature, subsonic air passes from the cathode, through a conical contraction, to the nozzle inlet, then expands through the nozzle to parallel, supersonic flow at the nozzle exit. The HEAT-H1 arc-heater configuration is shown schematically in Fig. 1.

This and the previous studies focus on the HEAT-H1 Mach 1.8 nozzle. This axisymmetric nozzle is 2.3 inches long, with a throat diameter of 0.9 inches. The material composition is copper-zirconium. The nozzle and surrounding cooling jacket are approximated in Fig. 2. Water flows through the cooling passage between the nozzle and cooling jacket, providing backside cooling for the nozzle wall. Away from

the end regions, heat transfer through the nozzle wall is approximately one-dimensional. Near the flanged

ends of the nozzle, the solid body heat transfer becomes axisymmetric in nature.
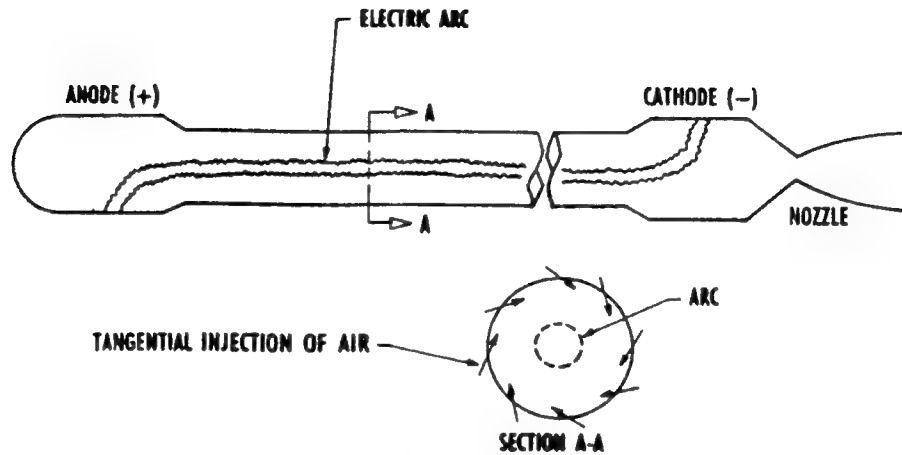

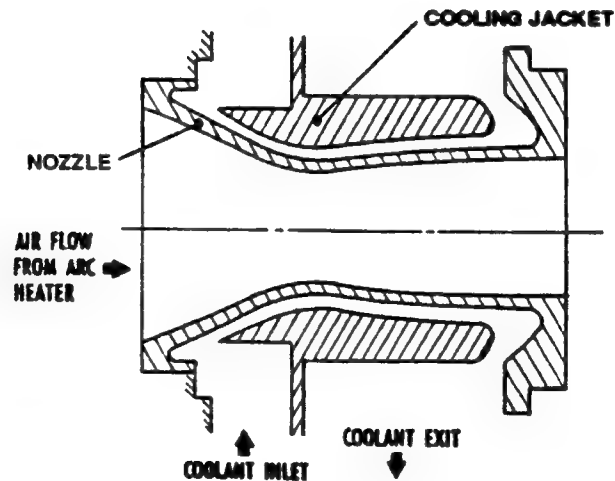Fig. 1. Schematic representation of the HEAT-H1 arc-heated wind tunnel.

Fig. 2. Idealization of the HEAT-H1 nozzle and cooling jacket.

Air flow over the nozzle wall is assumed steady, compressible, and turbulent, with a high

temperature boundary layer. Outside the boundary layer, flow is assumed to be steady, compressible,

nonuniform, dissociated, equilibrium air, with decreasing stagnation temperature approaching the wall. Transport properties for the high temperature, high pressure air are strong functions of temperature and weak functions of pressure.

Water flow through the cooling passage is assumed steady, incompressible, viscous, and turbulent. The water enters and exits the active cooling region in the subcooled state. Subcooled nucleate boiling is assumed to occur at the nozzle wall, while the outer cooling jacket wall is assumed adiabatic. Transport properties for the subcooled water are functions of temperature and pressure. Due to high flow convection rates, the momentum properties of the flow (density, velocity, and pressure) and the flow temperature distribution are decoupled.

Method of Analysis

In the original approach [4], the flow properties of water through the annular cooling passage were determined by application of the steady-state conservation laws to a one-dimensional control volume with area change. An elemental control volume was assumed with the known quantities,

A_{in} = inlet area,

A_{out} = exit area,

A_{wet} = wetted surface area,

D_{eq} = equivalent hydraulic diameter,

$\dot{m}$ = mass flow rate,

p_{in} = inlet static pressure,

μ = absolute viscosity,

ρ = constant mass density.

Then from conservation of mass in the control volume, the inlet and exit bulk velocities were

$$V_{in} = \frac{\dot{m}}{A_{in}\rho}, \text{ and } V_{out} = \frac{\dot{m}}{A_{out}\rho}.$$

The expression for conservation of momentum in the control volume was given by

$$p_{in}A_{in} - p_{out}A_{out} + \left(\frac{p_{in} + p_{out}}{2}\right)(A_{out} - A_{in}) - \tau A_{wet} = \dot{m}(V_{out} - V_{in}).$$

Here the turbulent wall shear stress τ was determined with Prandtl's universal law of friction for smooth pipes [5], and the known values D_{eq}, V_{in}, V_{out}, μ, and ρ. This left exit pressure as the only unknown in the momentum equation.

The improved approach uses the incompressible form of the full three-dimensional Navier-Stokes equations to obtain turbulent, steady-state velocity and pressure distributions of water through the cooling passage. The governing equations are formulated in the research flow solver INS3D-UP from NASA Ames Research Center [6]. This flow solver uses the pseudo-compressibility method, first introduced by Chorin [7], and successfully applied by Kwak et al. [8], Rogers and Kwak [9], and others.

Briefly summarizing this approach [10], the continuity equation for the incompressible Navier-Stokes equations reduces to

$$\bar{\nabla} \cdot \vec{V} = 0.$$

The steady-state momentum conservation equation, without body forces, becomes

$$\bar{\nabla} \cdot \left(\vec{V} \otimes \vec{V}\right) = \frac{1}{\rho} \bar{\nabla} \cdot \left(-p\bar{\bar{I}} + \bar{\bar{\tau}}\right),$$

where, $\vec{V}$ = Cartesian velocity vector $(u,v,w) = (V_1,V_2,V_3)$,

ρ = constant mass density,

p = static pressure,

$\bar{\bar{I}}$ = unit tensor,

$\bar{\bar{\tau}}$ = shear stress tensor = $\tau_{ij} = \mu\left(\dfrac{\partial V_j}{\partial x_i} + \dfrac{\partial V_i}{\partial x_j}\right)$,

μ = absolute viscosity.

The pseudo-compressibility method then replaces the continuity equation with the time-dependent

$$\frac{1}{\beta^2}\frac{\partial p}{\partial t} + \rho\bar{\nabla} \cdot \vec{V} = 0.$$

The parameter β represents a pseudo-speed of sound for the transformed system, with pseudo-pressure waves propagating at finite speed. Thereby, the elliptic governing equations have taken on a hyperbolic

9-7

character. The pseudo-speed of sound β must be adjusted for optimum convergence to the steady-state solution. The apparent transient behavior of the transformed system has no physical meaning until the divergence-free condition, and thus convergence, is reached. As implemented in INS3D-UP, an upwind differencing scheme based on flux-difference splitting is used to compute the convective terms, and second-order central differencing is used for the viscous terms [6, 9]. Also, the governing equations are nondimensionalized, and solved in a generalized curvilinear coordinate system.

An axisymmetric flow solution is obtained from INS3D-UP by solving the full three-dimensional problem on three azimuthally rotated planes. Axial symmetry boundary conditions are enforced on the lateral walls of the resulting wedge-shaped configuration, thus giving an axisymmetric solution on the center plane. For the HEAT-H1 problem, each of the two outer planes are rotated by 1° from the center plane.

The axisymmetric computational domain for the HEAT-H1 Mach 1.8 nozzle cooling passage is
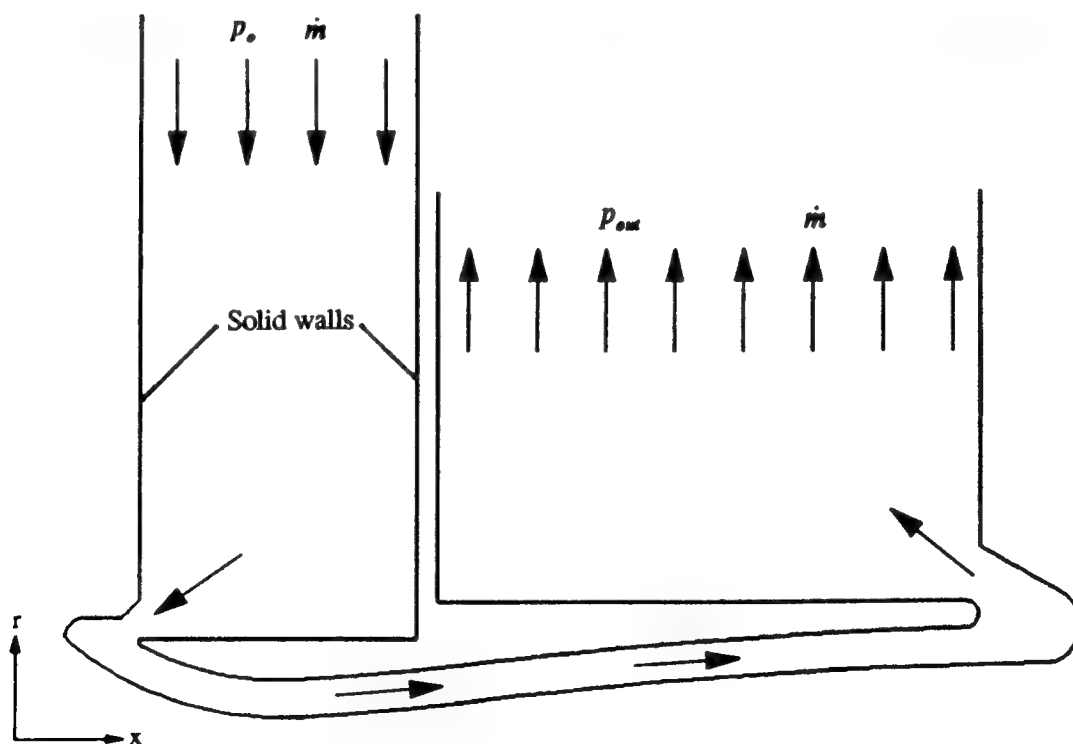
Fig. 3. Computational domain for the HEAT-H1 Mach 1.8 nozzle cooling passage.

9-8

shown in Fig. 3. No-slip velocity conditions are enforced at all solid boundaries. Flow enters the inflow boundary with a specified total pressure p_o, and exits the outflow boundary with a specified static pressure p_{out}. In the course of obtaining a flow solution, the pressure differential $(p_o - p_{out})$ must be adjusted to return the known system mass flow rate $\dot{m}$. Since experimental values for p_o and $\dot{m}$ are available, the pressure differential should be adjusted exclusively by varying p_{out}.

Experimentally measured values for p_o and $\dot{m}$ are shown in Table 1 for four heat load failure test cases culled from the HEAT-H1 Test Unit run history with the Mach 1.8 nozzle [2]. All four test cases have the identical value of 1,000 psia for the water inlet total pressure. Cases I and II have the identical value of 11.54 lb_m/s for water mass flow rate. The water mass flow rates for Cases III and IV are nearly equivalent. In the present analysis, results will be shown only for Case I inlet total pressure and mass flow rate conditions (also equivalent to Case II).

Table 1. Heat Load Structural Failure Test Cases for the HEAT-H1 Mach 1.8 Nozzle

Test Case Number	I	II	III	IV
Water inlet total pressure (psia)	1,000	1,000	1,000	1,000
Water mass flow rate (lb_m/s)	11.54	11.54	7.09	7.063
Water inlet temperature (R)	557	552	520	520
Water temperature rise (R)	25	27	34	36
Air total pressure (atm)	126.5	137	104.4	94.3
Air total temperature (R)	9,000	9,432	8,280	9,180
Air total enthalpy (Btu/lb_m)	3,480	3,750	3,062	3,644
Survival/Failure	Survived	Failed	Survived	Failed

The computational grid for the HEAT-H1 cooling passage contains 243 nodes tangential to the mean-flow direction, 61 nodes normal to the mean-flow direction, and two azimuthal planes rotated 1° from the central axisymmetric plane. In all, the computational grid contains 44,469 node points. Hyperbolic tangent node clustering is used in the normal mean-flow direction to capture boundary layer effects. In the tangential mean-flow direction, node points are tightly clustered in regions of anticipated large flow

gradients. For illustration, Fig. 4 shows a portion of the computational grid surrounding the first (upstream) flow turn. Tight clustering can be observed near the solid walls, and in the vicinity of the flow constriction. For clarity, node points in the normal mean-flow direction have been thinned by two-thirds.
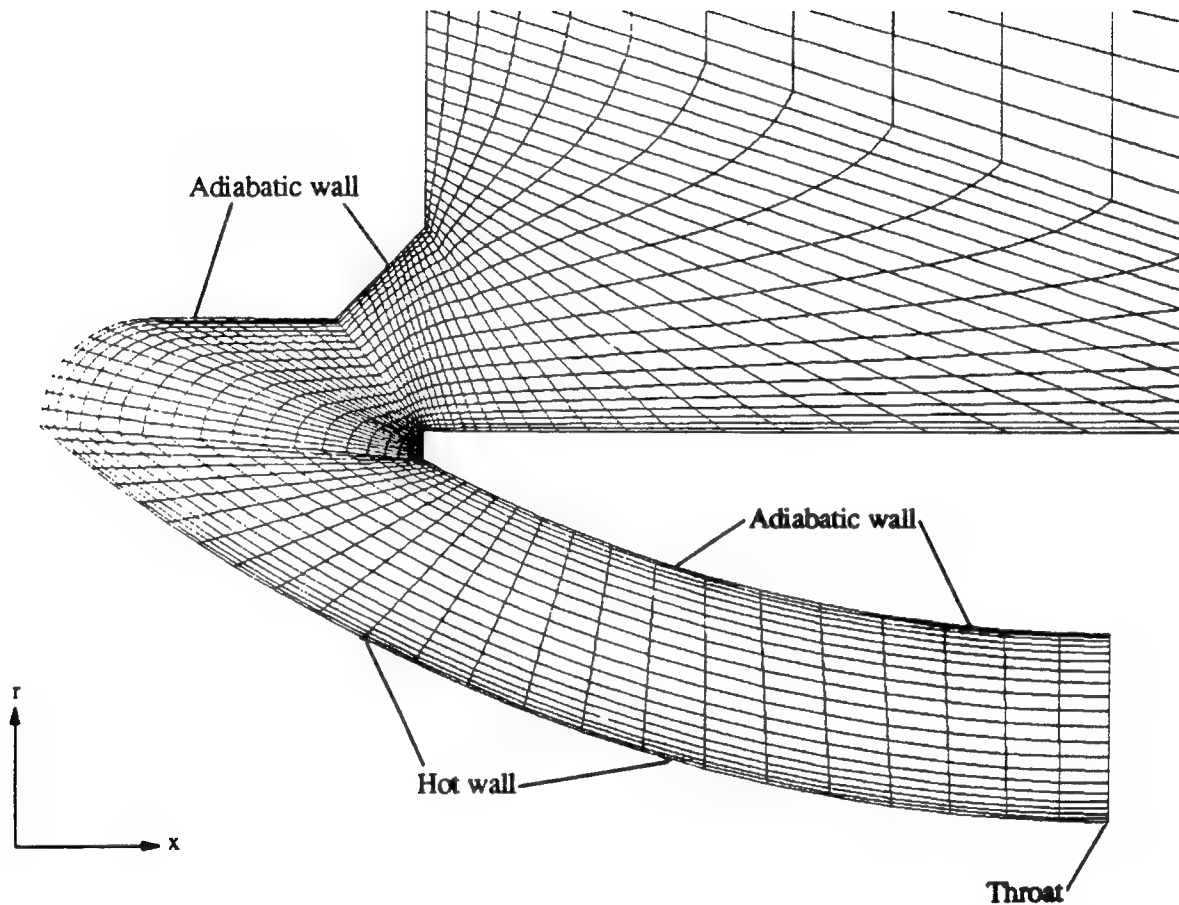

Fig. 4. Detail of the computational grid for the HEAT-H1 Mach 1.8 nozzle cooling passage.

Heat load deformation and failure in the HEAT-H1 nozzles has been observed in the region between the nozzle inlet and the nozzle throat. Nozzle wall failure has not been observed downstream of the nozzle throat. For this reason, the grid detail shown in Fig. 4, is also the area of primary interest for obtaining a flow solution. A recirculation region, or a nozzle-side (hot side) stagnation point in this area of the flow field might be indicative of observed heat load failures.

Results

INS3D-UP was used to obtain a converged flow solution for the HEAT-H1 Mach 1.8 nozzle cooling passage with Case I inlet conditions. For Case I, the inlet total pressure p_o was 1,000 psia, and the mass flow rate $\dot{m}$ was 11.54 lb_m/s. An outflow static pressure p_{out} of 832.5 psia was required to maintain the value of $\dot{m}$ to within 0.2%. The solution used a constant mass density ρ of 0.035727 lb_m/in^3, and a constant absolute viscosity μ of 3.4422×10^{-5} lb_m/in·s. A value of 10^7 for the parameter β (here, the square of the nondimensional pseudo-speed of sound) gave a stable, well-behaved convergence history for the solution. Convergence required 2,800 time steps, with a time step size of 10^{-3} (nondimensional units). Unsuccessful attempts at using larger time step sizes produced quick solution divergence.

Results from the flow solution indicate that large recirculation regions exist in the vicinity of the first flow turn. Velocity vectors scaled by magnitude in Fig. 5 show acceleration as the flow approaches
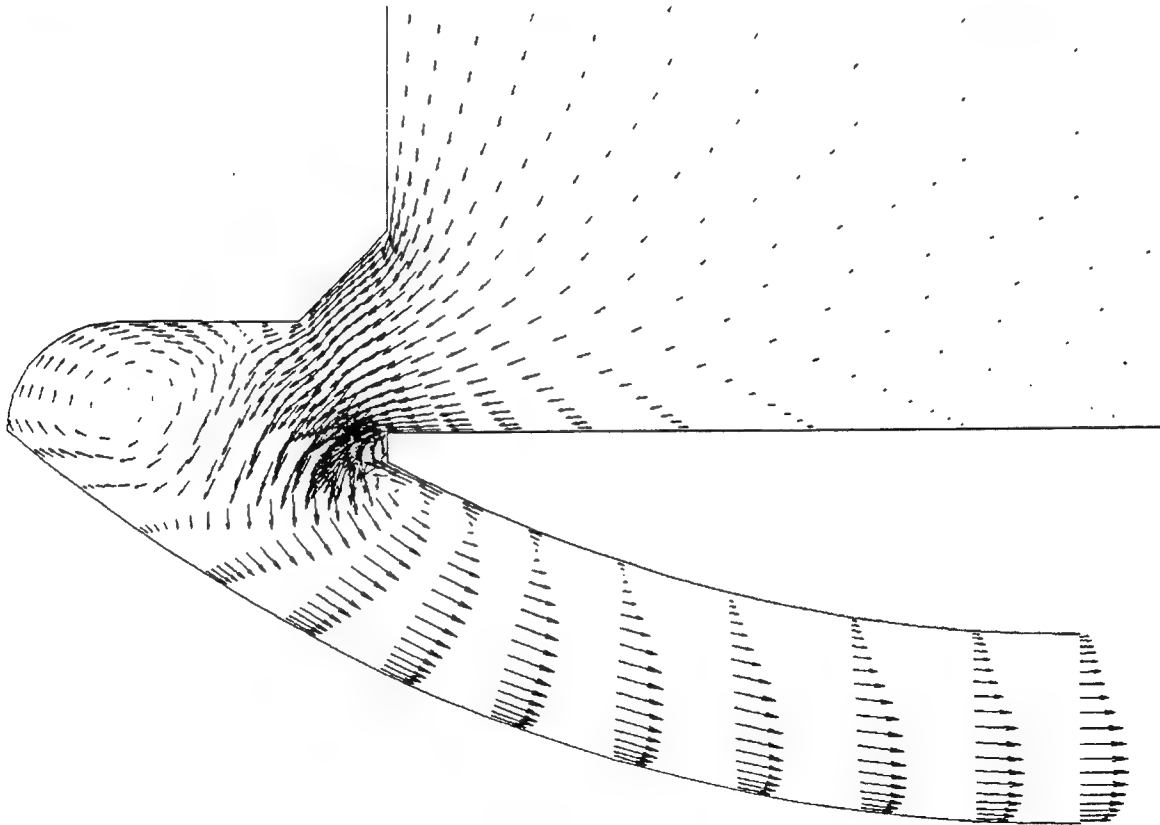
Fig. 5. Velocity vectors in the HEAT-H1 Mach 1.8 nozzle cooling passage (Case I).

the constriction at the beginning of the turn. As the flow negotiates the hairpin turn, two separation regions develop. The first separation region hugs the inside corner of the turn, and contributes to the elongated recirculation pattern carried partially downstream. The second separation region occurs on the outside edge of the turn, and leads to the large circular recirculation pattern.

Of particular interest, a stagnation point lies on the nozzle-side (hot side) of the cooling passage wall at the edge of the second separation region. A close-up view of this second region is shown in Fig. 6 with constant length vectors. The stagnation point appears as a confluence of velocity vectors on the lower wall. A complementary stagnation point exists on the upper, adiabatic wall.
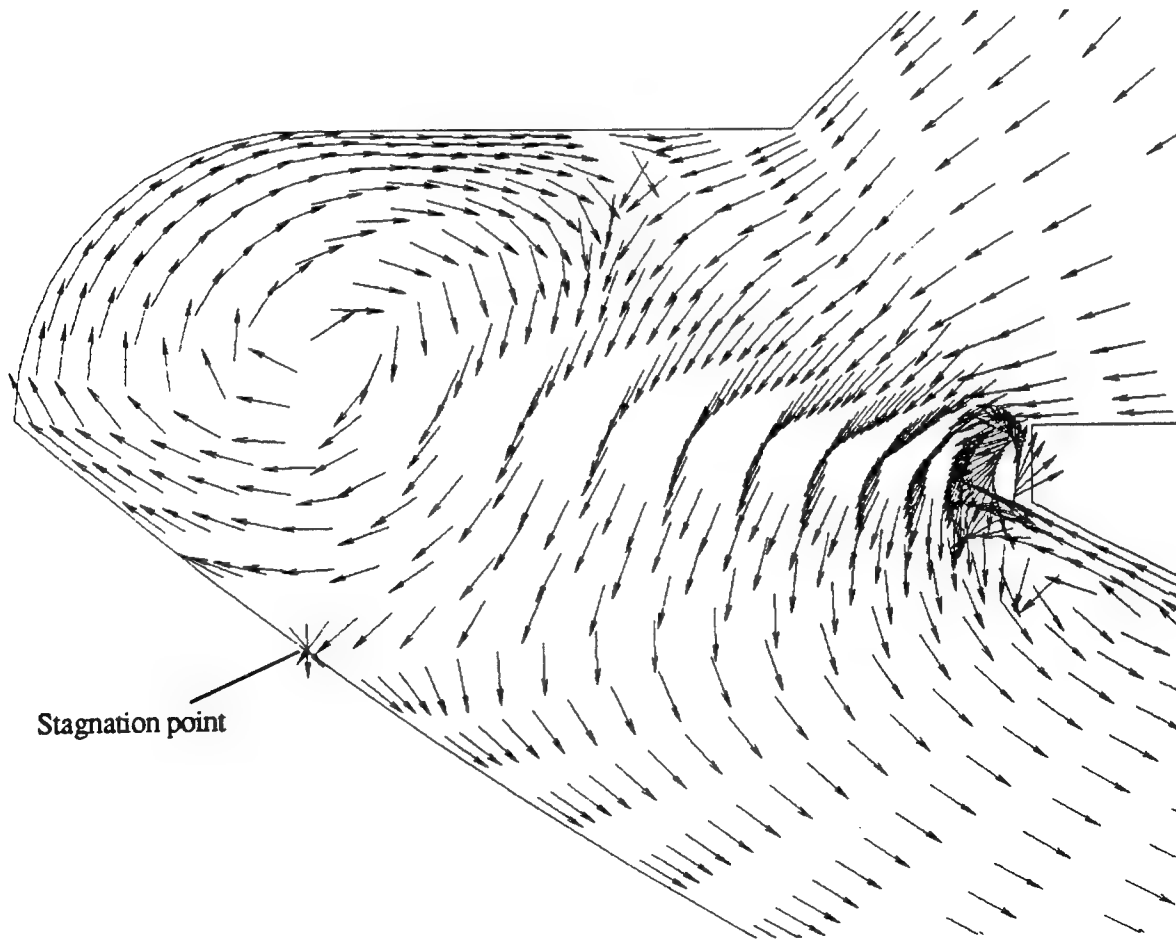
Fig. 6. Recirculation patterns in the HEAT-H1 Mach 1.8 nozzle cooling passage (Case I).

Pressure contours for the flow field correspond closely to the velocity solution. In Fig. 7, the stagnation region on the nozzle-side wall appears as a high pressure area of approximately 1,000 psia. Pressure values then drop to approximately 775 psia at the throat, as the flow accelerates downstream from the flow turn. The high pressure gradients across the constriction correspond to slope discontinuities in the computational grid. Also, the slope discontinuity at the sharp corner of the flow turn is most likely responsible for a small region of nonphysically negative static pressure (not apparent in the figure) near the corner. Any future work on flow analysis for the HEAT-H1 nozzle cooling passage must include improvement to the computational grid.
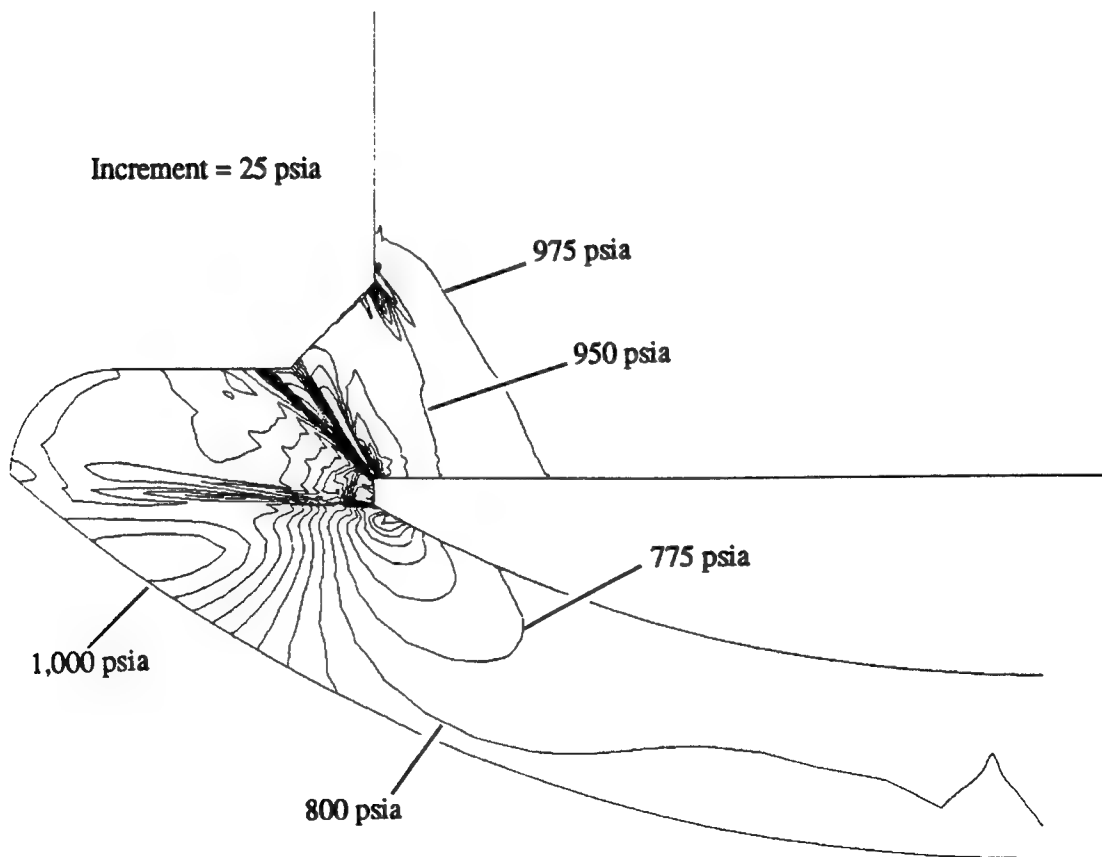
Fig. 7. Static pressure distribution in the HEAT-H1 Mach 1.8 nozzle cooling passage (Case I).

Conclusion

The existence of a recirculation region and a stagnation point on the hot wall reduces the efficiency of heat transfer from the wall. In the HEAT-H1 nozzle cooling passage, heat is swept away from the nozzle wall by continuously flowing cool water. If the water flow is allowed to recirculate in any single region of the wall, the local water temperature will rise, thus removing less heat from that wall region. Wall failure might occur when more heat is added to the air-side of the nozzle wall, than can be removed from the cooling passage side of the nozzle wall. The low or vanishing water velocities in the vicinity of the stagnation point will similarly reduce the amount of heat removed from that region of the wall. For these reasons, the cooling passage geometry should be redesigned to eliminate recirculation regions and stagnation points from the hot wall.

The HEAT-H1 nozzle cooling passage flow solution is one component of a conjugate heat transfer algorithm developed by the author to predict heat load failure in the HEAT-H1 Test Unit. Bulk flow velocities from the flow solution will be used with the energy conservation equation to obtain a bulk temperature distribution in the cooling passage. The energy solution does not need to be obtained simultaneously with the flow solution, because the incompressibility and high convection rates of the flow combine to decouple the continuity equation and momentum equations from the energy equation. Once determined, the bulk temperatures, along with bulk velocities and wall static pressures from the flow solution, will be used with Shah's correlation for subcooled nucleate boiling heat transfer [11, 12]. The subcooled nucleate boiling correlation will predict heat transfer coefficients along the hot wall of the HEAT-H1 nozzle cooling passage, for use in finding the axisymmetric conduction solution for the nozzle wall.

It is important to note that the flow solution obtained with INS3D-UP does not consider two-phase flow effects. Two-phase flow arises from the presence of subcooled nucleate boiling on the hot wall of the nozzle cooling passage. Water vapor bubbles would act to decrease the effective flow area, and thus might lead to higher flow velocities than would be expected in single-phase flow. Shah's correlation attempts to partially account for this effect.

9-14

Acknowledgments

This work was funded by AFOSR through the 1993 Summer Research Program at AEDC. The coordinating officer for the AFOSR Summer Research Program at AEDC was Maj. Hal Martin (USAF). The author wishes to acknowledge the contributions of Mr. F. L. Shope, for his technical expertise in the field of conjugate heat transfer, and Dr. B. R. Circelli for his knowledge of the INS3D-UP code and his advance work on the cooling passage flow solution.

References

[1]   *Test Facilities Handbook*, Arnold Engineering Development Center, Arnold AFB, Vol. 3, May 1992, p. 125.

[2]   Shope, F. L., "Conjugate Conduction/Convection/Nucleate-Boiling Heat Transfer with a High-Speed Boundary Layer," AIAA 3rd International Aerospace Planes Conference, Orlando, AIAA-91-5033, Dec. 1991.

[3]   Shope, F. L., "Conceptual Thermal Design of a 200-atm, Water-Cooled Arc Heater Nozzle," AIAA 28th Thermophysics Conference, Orlando, AIAA-93-2879, July 1993.

[4]   Weaver, M. A., "Multidimensional Conjugate Heat Transfer Analysis for the Arnold Engineering Development Center HEAT-H1 Test Unit Nozzle," 1992 AFOSR Summer Research Program Final Report, Sept. 1992.

[5]   Schlichting, H., *Boundary-Layer Theory*, McGraw-Hill Book Company, 1979, pp. 611-613.

[6]   Rogers, S. E., "Numerical Solution of the Incompressible Navier-Stokes Equations," NASA TM 102199, 1990.

[7]   Chorin, A. J., "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, vol. 2, 1967, pp. 12-26.

[8]   Kwak, D., Chang, J. L. C., Shanks, S. P., and Chakravarthy, S. R., "A Three-Dimensional Incompressible Navier-Stokes Solver Using Primitive Variables," *AIAA Journal*, vol. 24, no. 3, 1986, pp. 390-396.

[9]   Rogers, S. E., and Kwak, D., "Upwind Differencing Scheme for the Time-Accurate Incompressible

Navier-Stokes Equations," *AIAA Journal*, vol. 28, no. 2, 1990, pp. 253-262.

[10]   Hirsch, C., *Numerical Computation of Internal and External Flows, Volume 2: Computational Methods for Inviscid and Viscous Flows*, John Wiley & Sons Ltd., 1990, pp. 654-656.

[11]   Shah, M. M., "A General Correlation for Heat Transfer during Subcooled Boiling in Pipes and Annuli," *ASHRAE Transactions*, vol. 83, pt. 1, 1977, pp. 202-217.

[12]   Shah, M. M., "Generalized Prediction of Heat Transfer during Subcooled Boiling in Annuli," *Heat Transfer Engineering*, vol. 4, no. 1, Jan.-March 1983, pp. 24-31.

Application of the Volterra Functional Series
Approach for Understanding Control of Flow Separation
Downstream of a Pitching Airfoil

Charles Pezeshki
Assistant Professor
Department of Mechanical and Materials Engineering

and
✳ Wyatt O. Davis
Graduate Student

Washington State University
Pullman, WA 99164-2920

Application of the Volterra Functional Series Approach for Understanding Control of Flow Separation Downstream of a Pitching Airfoil

Charles Pezeshki
Assistant Professor
Department of Mechanical and Materials Engineering
Washington State University

and

Wyatt O. Davis
Graduate Student

Abstract

A combination experimental/numerical investigation of flow separation for a static airfoil has been conducted using Higher-Order Spectral Analysis (HOS) and the Volterra functional series approach. Input and output time series were recorded downstream of the airfoil at certain points in the flow to investigate and quantize the natural energy transfer mechanisms present. Time series were recorded for two angles of attack, and hot-film anemometers were used to record input and output flow velocities downstream from the airfoil at various fractions of the chord length downstream. Linear and quadratic transfer functions were calculated for these trajectories. These functions were then examined for evidence of natural energy transfer mechanisms in the flow.

Following this analysis of the unforced flow, a study was conducted using pulsed air as a mechanism for control of flow separation. Linear and quadratic transfer functions were calculated for these test cases, and insights were obtained on the mechanism by which the angle of incipient flow separation is increased.

Introduction

A large body of previous work has been done on controlling growth of a two-dimensional mixing layer. In particular, the work of Oster and Wygnanski [OSTE] examined the use of a small flap to periodically force the flow and alter the growth of the flow instability downstream of the plate. Ho and Huerre [HO] discussed the use of acoustic as well as mechanical forcing for control of the mixing layer. More relevant to the problem of flow separation downstream of a static airfoil, Lovato [LOVA] investigated active control of the shear layer on an airfoil using external acoustic forcing and internal-tangential pulsed air control. Their results showed that the airfoil shear layer responds to the active forcing akin to a free shear layer.

An extensive amount of research has been performed concerning the analysis of fluid flows using Higher-Order Spectral analysis (HOS). Of particular interest to the research discussed below is the work of Miksad, Jones et. al [MIKS]. They studied the transition of a laminar two-dimensional wake to establish the role of amplitude and phase modulations in the spectral-broadening and energy-redistribution process. Later work by Ritz, Powers and Miksad et. al. [RITZ] concerned the spectral dynamics of transitioning flow in the wake of a flat plate. For this work, quadratic transfer functions from a Volterra functional approximation were calculated to examine the nonlinear spectral dynamics occurring between two sensor points. Finally the work by Hajj, Miksad and Powers [HAJJ] involving quantification of the possible nonlinear and parametric resonance mechanisms associated with the transition to turbulence in plane mixing layers was most influential in the following research. In this paper, HOS analysis techniques were used to investigate both nonlinear and parametric transfers of energy to lower frequencies.

The work performed by the principal investigator this summer of 1993 builds on the above work by examining the flowfield downstream from a static airfoil fixed at an angle of attack. Many comparisons have been formed with the flow downstream of an airfoil and both a mixing and shear layer [LOVA]. Hot film anemometry was used to record two time series simultaneously downstream from the airfoil at various relative positions to the airfoil. Quadratic transfer

functions were then calculated for the time series data using the approach developed by [RITZ], and these were analyzed to confirm fundamental flow frequencies and basic energy transfer mechanisms in the flow. In addition, a brief study was done on the energy transfer mechanisms involved in the dynamically pitching airfoil. A similar analysis was conducted as for the static case, with the hot film anemometers held at a fixed position relative to the pitching airfoil. This work will lead to future study of this important problem.

Experimental Set-up

The hot film anemometry experiments were conducted in the Frank J. Seiler Laboratory's open return, low speed wind tunnel at the U.S. Air Force Academy. The .91m x .91 m test section is designed for use with flow visualization and flow sensor measurements, and has free stream turbulence intensity levels below 0.5% A NACA-0015 airfoil with a 15.2 cm chord and a 61.0 cm span was used for all tests. Chord Reynolds numbers were at 24,000 for the hot-film data. The two time series were recorded using dual 60 mil hot-film probes in conjunction with a TSI Model 1054A anemometer. The overheat ratio was set at 30%. All data acquisition utilized a MassComp MC-5500 micro-computer. The flow was sampled at a frequency of 80 Hz for use in the HOS and transfer function analysis. The signals were all filtered with a low-pass filter at 300 Hz.

Flow modification was accomplished individually by internal tangential-pulsed air blowing. Pulsed air was directed through a downstream facing slot located on the upper airfoil surface. The slot was 1.6 mm wide and spanned the middle 15.0 cm of the airfoil, or 25% of the entire span, located at the leading edge of the airfoil. The system employed two solenoid valves capable of pulsing between zero and 45 Hz. The valves were controlled through a timing circuit using a square wave generator. Compressed air was supplied at 30 psig through tygon tubing into each end of the slot, resulting in a peak injection velocity of 8.5 m/sec. The blowing momentum coefficient, C_μ, was 0.07 for a chord Reynolds number of 24,000.

Transfer Function Calculations

The method for calculating nonlinear transfer functions to second order is summarized in [RITZ] and [HAJJ]. The general idea behind such work is that a nonlinear response of a dynamical system can be condensed, up to second order, into additive responses consisting of a linear frequency response, $L(f_m)$, plus a quadratic frequency response, Q, as given by

$$Y(f_m) = L(f_m)X(f_m) + \sum_{f_i \pm f_j} \sum Q_{f_i,f_j}^{f_m} X(f_i)X(f_j) \quad f_m = f_i \pm f_j$$

Basically, it involves solving two moment equations in the frequency domain, given by [RITZ] as

$$E\left[Y(f_m)X^*(f_m)\right] = L(f_m)E\left[X(f_m)X^*(f_m)\right] + \sum_{f_i \pm f_j} \sum Q_{f_i,f_j}^{f_m} E\left[X(f_i)X(f_j)X^*(f_m)\right]$$

and

$$E\left[Y(f_m)X^*(f_k)X^*(f_l)\right] = L(f_m)E\left[X(f_m)X^*(f_k)X^*(f_l)\right] + \sum_{f_i \pm f_j} \sum Q_{f_i,f_j}^{f_m} E\left[X(f_i)X(f_j)X^*(f_k)X^*(f_l)\right]$$

where

$$f_m = f_i \pm f_j = f_k \pm f_l$$

where f_m is the output frequency of the model, E is the expectation operator and $X(f_j)$ and $Y(f_j)$ are the Fourier transforms of the input and the output respectively. Importantly, these two equations allow one to express both transfer functions in terms of the various HOS moments that are given by the upstream and downstream sensors of the probe [HAJJ]. For example, the LHS of the first equation is the cross-spectrum, the RHS of the first equation associated with the linear transfer function is the power spectrum, and with the quadratic transfer function, the auto-bispectrum. For the second equation, the LHS is the cross-bispectrum, the term associated with the linear transfer function is the auto-bispectrum, and the term associated with the quadratic transfer function is the fourth-order spectral moment. Both these equations generate a square matrix with enough equations to solve for all the unknowns in both the linear and quadratic transfer functions.

This summer, a large program was written that would assemble these two equations into a matrix and solve for both the linear and quadratic transfer

functions. The name of this program is SKETCHY. SKETCHY initially reads in the input and output time series recorded from the hot-film anemometry probe. Following this, the time series is broken up into a number of blocks, and the various averaged spectral moments are calculated, such as the auto- and cross-spectrum, the auto- and cross bispectrum and the fourth-order spectral moment. These moments are then used to construct a large two-dimensional square matrix associated with the unknowns, along with a column vector for the various terms in the transfer functions. This is solved using a lower/upper triangular solver. The unknowns are then reassembled into appropriate matrices representing the linear and the quadratic transfer functions. The magnitudes of these complex arrays are taken and the results are written to output files, which are displayed using a plotting application developed by W. Davis using the graphics language IDL (Interactive Display Language) by Research Systems Incorporated. To further understanding of the quadratic transfer function results, a non-dimensional quadratic transfer function is also calculated, defined as [HAJJ]

$$Q'^2(f_i,f_j) = E\left[\left|Q_{f_i,f_j}^{f_m}\right|^2\left(\left|X(f_i)X(f_j)\right|^2\right)^{\frac{1}{2}}\right], \quad f_m = f_i + f_j$$

This nondimensionalized quadratic transfer function shows the magnitude of the energy transferred due to a given interaction, combining affinity for a given three-wave interaction to exchange energy multiplied by the actual energy present to be transferred.

The program was verified for correctness of results by experimenting with sample cases where the linear and quadratic transfer functions were known. In particular, two input-output combinations, one to test for a pure linear response, and the other to test for a pure quadratic response were analyzed, and both cases yielded the correct answer. However, preliminary examination of the matrix used to solve for the linear and quadratic transfer functions indicate that the matrix is poorly conditioned. One of the highest priorities upon continuance of this work will be to use a sparse matrix solver designed for poorly conditioned matrices, as well as implement the entire code in double precision. Because of the size of the matrices manipulated, this will involve use of a supercomputer.

Results and Discussion

Quadratic transfer functions were calculated using 16384 points collected by the Masscomp computer, subdivided into 64 point blocks for the averaging techniques. The sampling rate was chosen to be 80Hz, well above the necessary Nyquist frequency needed to examine the higher frequency content. The lowest frequency available for examination was 1.25Hz. This was deemed adequate for a preliminary study of the phenomena; however, the final study to be conducted this winter will use more data and a 128 point block size.

For the static case, hot film pair measurements were made in a variety of positions downstream of the airfoil. Two angles of attack were chosen: 15° and 20°. Six cases were analyzed for the 15° angle of attack and seven cases for the 20° angle of attack downstream of the airfoil.

Figure 1 is a representative plot of the nondimensionalized quadratic transfer function at 15° angle of attack for the unforced case. The transfer function shows that the flow is self-organizing, with energy from all frequencies in the input being coupled to, or transferring energy to 5 Hz, the lowest subharmonic present in the flow, 20 Hz, the main harmonic, and 30 Hz, the first superharmonic. The primary frequency triads present in this plot are (15Hz, 5Hz, 20Hz), (10Hz, -5Hz, 5Hz), and (20Hz, 10Hz, 30Hz).

Figure 2 is the plot of the nondimensionalized quadratic transfer function at 20° angle of attack. The flow is also self-organizing, transferring energy primarily to approximately 4 Hz, the lowest subharmonic, and 30 Hz, the first superharmonic in the flow, with large energy coupling mechanisms at (10Hz,-6Hz,4Hz) and (20Hz,10Hz,30Hz).

Because of the large amount of coupling observed between for the frequency values of 5Hz, 10Hz and 20Hz in the natural flow case, indicating strong flow organization because of these frequencies, a pulsed air control at these frequencies was used to control flow separation at 20° angle of attack. For 5Hz pulsed air frequency, the flow quickly becomes organized and periodic, with

strong frequency content at 10Hz, 15Hz, and 25Hz. Figure 3 shows the nondimensionalized quadratic transfer function for a typical trial. Strong coupling exists between 5Hz in the input and 10Hz in the output, from frequency triad (5Hz, 5Hz, 10Hz). In addition, the pulsed air serves as a strong organizer of the flow, as all energy from all frequencies is directed into the 10Hz, 15Hz and 25 Hz frequencies in the output.

Conclusions

Preliminary work concerning flow control for separation from a static airfoil mounted at given angles of attack was completed this summer by the P.I. The Volterra functional series/ quadratic transfer function calculation method was implemented and used to obtain results for flow control. Initial results are promising; fundamental flow frequencies are successfully identified, and energy transfer mechanisms in the flow are highlighted. Future work will involve examining the dynamic pitching airfoil for energy transfer mechanisms so that a control scheme can be proposed for delaying the onset of dynamic stall, and enhancing airfoil performance in the post-stall regime.

Bibliography

[HAJJ] Hajj, M.R., Miksad, R.W., and Powers, E. J., (1992) "Subharmonic Growth by Parametric Resonance", J. Fluid Mech., vol. 236, pp. 385-413.

[HO] Ho, C.M. and Huerre, P. (1984) "Perturbed Free Shear Layers," Ann. Rev. Fluid Mech., Vol. 119, pp. 443-473.

[LOVA] Lovato, J.A., (1992) "Active Control of the Separation Region on a Two-Dimensional Airfoil", FJSRL TR-92-0001.

[MIKS] Miksad, R.W., Jones, F.L. Powers, E.J., Kim, Y.C., and Khadra, L., (1982) "Experiments on the Role of Amplitude and Phase Modulations during Transition to Turbulence", J. Fluid Mech., Vol. 123, pp. 1-29.

[OSTE] Oster, D. and Wygnanski, I. (1982) "The Forced Mixing Layer Between Parallel Streams," J. Fluid Mech., Vol. 123, pp 91-130.

[RITZ] Ritz, C. P., Powers, E.J., Miksad, R.W., and Solis, R.W., (1988) "Nonlinear Spectral Dynamics of a Transitioning Flow", Phys. Fluids, Vol. 31(12), pp. 3577-3588.
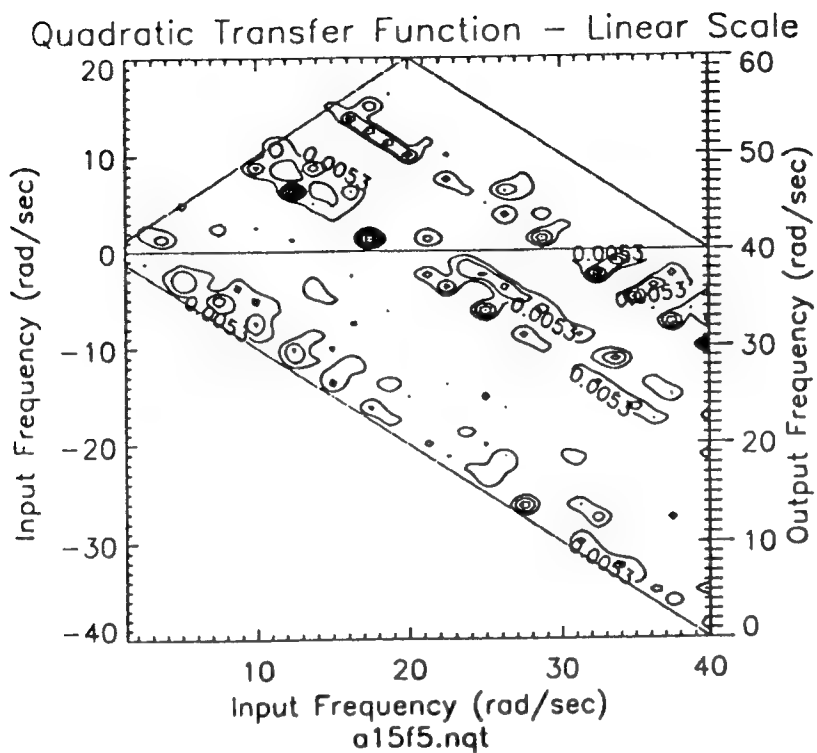
Figure 1. Nondimensionalized quadratic transfer function, angle of attact 15°, unforced case.
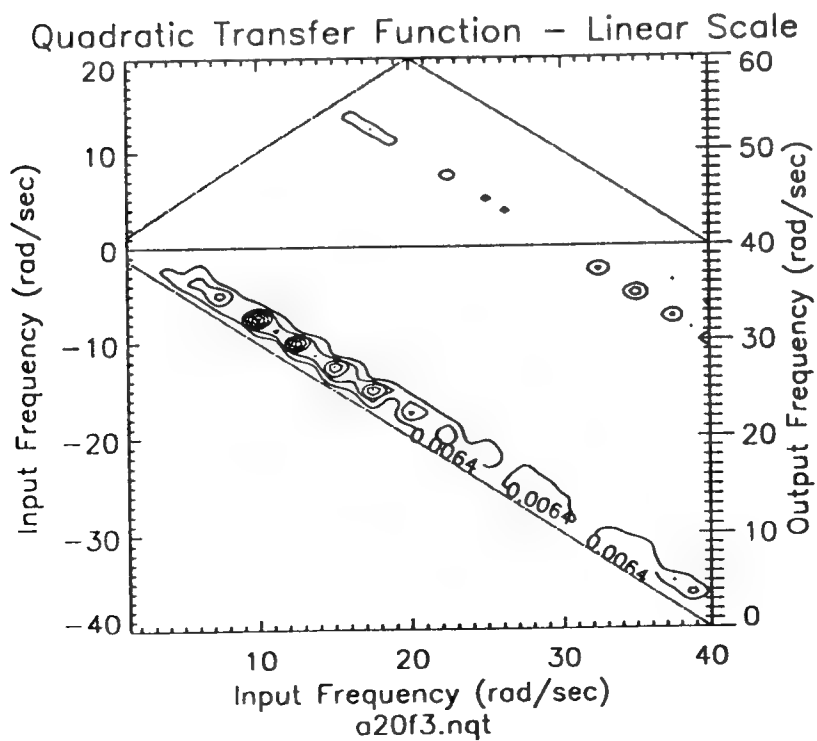

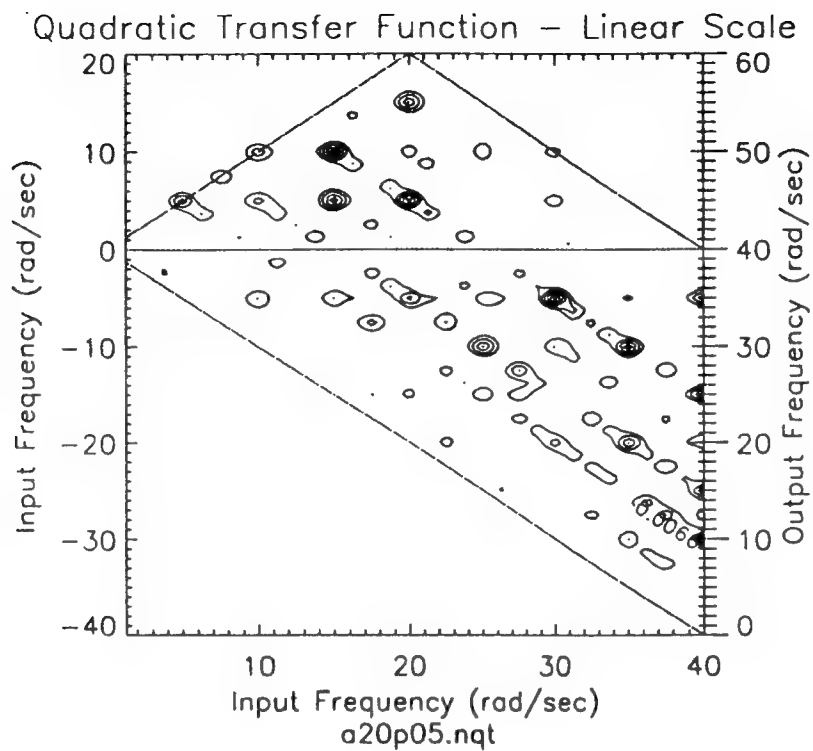Figure 2. Nondimensionalized quadratic transfer function, angle of attack 20°

10-10

Figure 3. Nondimensionalized quadratic transfer function, angle of attact 20°, pulsed air forcing at 5Hz.

Alan DeVilbiss's report not available at time of publication.

ACTIVE CONTROL DESIGN FOR THE FJSRL SLEWING PIEZOELECTRIC LAMINATE BEAM

Travis V. DuBois

Graduate Student

Department of Mechanical Engineering and Mechanics

Old Dominion University

Norfolk, Virginia 23529-0247

ACTIVE CONTROL DESIGN FOR THE FJSRL SLEWING

PIEZOELECTRIC LAMINATE BEAM

Travis V. DuBois
Department of Mechanical Engineering and Mechanics
Old Dominion University
Norfolk, Virginia 23529-0247, USA
Phone: (804) 683-3736
Email: dubois@mem.odu.edu

Abstract

The research performed pertains to the development and testing of synergistic active and passive control designs for the piezoelectric laminate beam experiment at FJSRL. Under the AFOSR 1992 Summer Research Program, a multi-input, multi-output transfer matrix model for a slewing beam system with piezoelectric actuators and sensors, was experimentally verified. The piezoelectric beam system at FJSRL was used for this work.

The transfer functions previously established for piezoelectric laminate beam experiment were used to develop and experimentally validate a simultaneously optimal active and passive damping design for the experimental system. However only the optimal active damping was verified here. A preliminary damping design has already been developed and tested at FJSRL. In this study, IMSC, (Independent Modal Space Control) was used for an optimal active control. An optimal active and passive design synergism has previously been developed. The work reported here supports ongoing the progress of contract #RDL 93-156, toward verification of both the active and passive control measures together.

ACTIVE CONTROL DESIGN FOR THE FJSRL SLEWING PIEZOELECTRIC LAMINATE BEAM

Travis DuBois

1 Introduction

It is well known that the achievable performance of active control designs can be limited by a number of practical factors, including spillover effects [1] which could lead to instability. Indeed, in initial experiments with the JPL precision truss, Fanson, et.al. [2] found that simple controllers successfully attenuated lower frequency modes but inadvertently destabilized higher modes. The augmentation of active control with passive damping provides a practical solution [3] to such problems. Viscoelastic passive damping treatments can be designed to perform well in a prescribed frequency range and are particularly well suited for the control of higher modes of structural vibration. Moreover, active control of higher modes is seldom required for purposes of, for example, shape control. Rather, enhanced damping of the higher modes is the desired property. Active control measures, are normally viewed as being well suited only for lower frequency structural modes. due to actuator bandwidth and computational speed limitations. Considered together, these facts suggest that a synergistic approach employing active control for lower frequency modes and passive control for higher frequency modes deserves serious consideration. Passive damping can enhance the performance of active controllers while easing the burden of active control and providing enhanced system stability [3]. Desirable properties of passive control approaches include simplicity, reliability, low cost and light weight. The goal of this research was to develop and experimentally validate a simultaneously optimal active and passive damping design for the experimental system. The control designs for the beam experiment will be based on the models developed and verified during the summer of 1992 at FJSRL. The initial approach to optimal active/passive control synergism will employ the Independent Modal Space Control (IMSC) [4], [5], [6] method for active control. IMSC was chosen due to its simplicity and the intuitive feel that it lends to the active control design problem.

2 Piezoelectric Laminate Electromechanical Relationships

A typical section of the beam/piezoelectric layer laminate has the top piezoelectric layer, the base structure and the lower piezoelectric layer with the subscripts a, b, and c respectively.

In the present development, the a layer will serve as actuator and the c layer as sensor. A voltage V_a

applied across the actuating layer a induces a longitudinal stress σ_f given by:

$$\sigma_f(x,t) = E_a \frac{d_{31}}{t_a} V_a(x,t) \tag{1}$$

where d_{31} is the electric charge constant of the film (m/v), and E_a is the Young's modulus of the film (N/m^2). This stress in turn generates a bending moment M_{f_a} about the composite system's neutral axis, given by [7]:

$$M_{f_a}(x,t) = \int_{\frac{t_b}{2}}^{\frac{t_b}{2}+t_a} \sigma_f wy \, dy = \frac{1}{2} E_a d_{31} b(t_a + t_b) V_a(x,t) = C_a V_a(x,t) \tag{2}$$

where $C_a = \frac{1}{2} E_a d_{31} w(t_a + t_b)$ is a geometric constant, the form of which is determined by the geometry of the composite beam-actuator-sensor system. The constant changes if, for example, both the a and c layers are used as actuators with opposite polarity, then $C_a = E_a d_{31} w(t_a + t_b)$. It is also worth noting that the expression for C_a is significantly simplified by the fact that the laminated beam's neutral axis coincides with its center due to the symmetry of the added piezoelectric elements.

Next an expression is obtained for the voltage across the sensing layer. Note that the strain ϵ_c is related to the curvature of the·beam in the following way:

$$\epsilon_c(x,t) = -\left(\frac{t_b}{2} + t_a\right) \frac{\partial^2 y}{\partial x^2} \tag{3}$$

Due to the piezoelectric effect this strain gives rise to a charge distribution per unit area, $q(x,t)$, which is given by:

$$q(x,t) = \left(\frac{k_{31}^2}{g_{31}}\right) \epsilon_c(x,t) \tag{4}$$

where k_{31} is the piezoelectric electromagnetic coupling constant and g_{31} is the piezoelectric stress constant. The total charge developed on the sensing layer is obtained by integrating $q(x,t)$ over the entire surface area of the piezo element,

$$Q(t) = \int_{x_1}^{x_2} wq(x,t) dx = -w\left(\frac{t_b}{2} + t_a\right) \frac{k_{31}^2}{g_{31}} \left.\frac{\partial y(x,t)}{\partial x}\right|_{x_1}^{x_2} \tag{5}$$

The piezoelectric material is similar to an electric capacitor and the voltage across the two layers is given by the following formula:

$$V_s(t) = \frac{Q(t)}{Cw(x_2 - x_1)} = C_s \left.\frac{\partial y(x,t)}{\partial x}\right|_{x_1}^{x_2} \tag{6}$$

where C_s is a constant incorporating various structural and piezoelectric constants in (5) and (6), C is the capacitance per unit area, and $w(x_2 - x_1)$ is the piezoelectric element's surface area. In the next section a dynamic modelling method for piezoelectric laminate beams is presented using the relationships given in this section.

3   Dynamic Modeling

The system under consideration is a generic model representing slewing flexible beams with piezoelectric actuators and sensors. The beam is mounted on rotary base, actuated by a torque motor, and rotating in the horizontal plane. The piezoelectric patches are attached to both sides of the beam. In this development, one side acts as an actuator and the other acts as a sensor. Thus the system inputs consist of voltage V_a applied to the actuating layer and the torque T applied by the torque motor. Outputs include the voltage V_s sensed by the sensing layer, motor hub angle θ and beam tip position Y_{tip}.

Using the usual assumptions for technical beam theory, the differential equation of motion for the composite beam-film system can be expressed as a Bernoulli-Euler beam equation with an additional term due to the actuating layer [8]. The equation is given as

$$\frac{\partial^2}{\partial x^2}\left[EI\frac{\partial^2 y(x,t)}{\partial x^2} - C_a V_a(x,t)\right] + \rho A \frac{\partial^2 y(x,t)}{\partial t^2} = 0 \tag{7}$$

where E and I represent respectively the effective Young's modulus and area moment of inertia for the composite system. Notice that if the piezoelectric laminates do not cover the entire beam surface, then both EI and ρA are functions of x. In the laminated portions of the beam, the product EI is given by $EI = E_a I_a + E_b I_b + E_c I_c$, and the product of the effective mass density ρ with the composite system's cross sectional area A is $\rho A = \rho_a A_a + \rho_b A_b + \rho_c A_c$. Since the piezoelectric layers are often thin by comparison to the base structure, the present analysis is simplified by assuming that EI and ρA are uniform over the length of the beam. The boundary conditions are:

$$y(0,t) = 0 \tag{8}$$

$$EI\frac{\partial^2 y(0,t)}{\partial x^2} - I_h \frac{\partial^3 y(0,t)}{\partial t^2 \partial x} + T(t) = 0 \tag{9}$$

$$EI\frac{\partial^2 y(l,t)}{\partial x^2} + I_t \frac{\partial^3 y(l,t)}{\partial t^2 \partial x} = 0 \tag{10}$$

$$EI\frac{\partial^3 y(l,t)}{\partial x^3} - M_t \frac{\partial^2 y(l,t)}{\partial t^2} = 0 \tag{11}$$

To allow conversion to transfer functions, the differential equation (7) and boundary conditions (8)–(11) are Laplace transformed. Letting $\beta^4 = -\rho A s^2/EI$, (7) becomes

$$Y''''(x,s) - \beta^4 Y(x,s) = \frac{C_a V_a''(x,s)}{EI} \tag{12}$$

with boundary conditions:

$$Y(0,s) = 0 \tag{13}$$

$$EIY''(0,s) - I_h s^2 Y'(0,s) + T(s) = 0 \tag{14}$$

$$EIY''(l,s) + I_t s^2 Y'(l,s) = 0 \tag{15}$$

$$EIY''''(l, s) - M_t s^2 Y(l, s) = 0 \tag{16}$$

where the primes indicate spatial derivatives. Together, equations (12)–(16) constitute a linear ordinary differential equation in x with mixed boundary conditions — two at $x = 0$ and two at $x = l$. The differential equation (12) is a Laplace transformed version of a classical Bernoulli-Euler beam equation with the second spatial derivative of $V_a(x, s)$ as a forcing input. Notice that when the spatial distribution of the actuating layer is uniform with respect to x, then the voltage term has a non-zero contribution only at the boundaries of the actuating layers. The actuating voltage V_a is constant in the interval $x_1 < x < x_2$, but undergoes a step change at each of the boundaries of this interval. Thus, the second spatial derivative of the actuating voltage function yields,

$$V_a''(x, s) = [\delta'(x - x_1) - \delta'(x - x_2)]V_a(s) \tag{17}$$

where $\delta'(\cdot)$, the spatial derivative of the dirac delta, represents the unit dipole function. It can be seen that, in effect, the actuating layer applies moments to the beam at its endpoints x_1 and x_2. For a system with N piezoelectric actuator/sensor pairs a state-space type representation can be formed:

$$\begin{bmatrix} Y'(x, s) \\ Y''(x, s) \\ Y'''(x, s) \\ Y''''(x, s) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \beta^4 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Y(x, s) \\ Y'(x, s) \\ Y''(x, s) \\ Y'''(x, s) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \frac{C_a V_a(s)}{EI} \sum_{i=1}^{2N} \delta'(x - x_i)(-1)^{i+1} \tag{18}$$

Defining

$$Z(x, s) \triangleq [Y(x, s) \; Y'(x, s) \; Y''(x, s) \; Y'''(x, s)]^T$$

and

$$U(x, s) \triangleq \frac{C_a V_a(s)}{EI} \sum_{i=1}^{2N} \delta'(x - x_i)(-1)^{i+1}$$

equation (18) can be written in compact form with appropriate definitions of A and B as follows:

$$Z'(x, s) = AZ(x, s) + BU(x, s) \tag{19}$$

Then using the fact [9] that,

$$\int_{-\infty}^{\infty} \delta^n(x)\phi(x)\,dx = (-1)^n \phi^n(0)$$

equation (19) can be expressed for $N = 1$, (i.e. only one piezoelectric actuator/sensor pair) as:

$$Z(l, s) = e^{Al} Z(0, s) + [Ae^{A(l-x_1)}B - Ae^{A(l-x_2)}B]\frac{C_a V_a(s)}{EI} \tag{20}$$

Now combining (20) with the boundary conditions (13)–(16) yields 8 linear algebraic equations in 8 unknowns, which can be solved simultaneously for the elements of $Z(0, s)$ and $Z(l, s)$. Once the end conditions $Z(0, s)$ and $Z(l, s)$ are known a general expression for $Z(x, s)$ can be written as follows:

$$Z(x, s) = \begin{cases} e^{Ax} Z(0, s), & 0 \leq x \leq x_1 \\ e^{Ax} Z(0, s) + Ae^{A(x-x_1)}B\frac{C_a V_a(s)}{EI}, & x_1 \leq x \leq x_2 \\ e^{Ax} Z(0, s) + A[e^{A(x-x_1)} - e^{A(x-x_2)}]B\frac{C_a V_a(s)}{EI}, & x_2 \leq x \leq l \end{cases} \tag{21}$$

The matrix exponential function e^{Ax} corresponding to (18) can be expressed as

$$e^{Ax} = \begin{bmatrix} f'''(x) & f''(x) & f'(x) & f(x) \\ \beta^4 f(x) & f'''(x) & f''(x) & f'(x) \\ \beta^4 f'(x) & \beta^4 f(x) & f'''(x) & f''(x) \\ \beta^4 f''(x) & \beta^4 f'(x) & \beta^4 f(x) & f'''(x) \end{bmatrix} \qquad (22)$$

where, $f(x) = \frac{1}{2\beta^3}[\sinh(\beta x) - \sin(\beta x)]$.

The solution expressed above by (21) has been derived for a single piezoelectric element, however, because of the system's linearity, superposition allows the same analysis to be applied to systems with any number of piezoelectric elements. In the analysis which follows, transcendental expressions are developed for the transfer functions of interest. To reduce the complexity of the expressions, the tip mass M_t and inertia I_t are set to zero for the remainder of the paper.

4   Transfer Functions

The system under consideration has two discrete inputs, V_a and T, and three outputs, Y_{tip}, θ and V_s. Thus, the transfer matrix can be expressed:

$$\begin{bmatrix} \theta(s) \\ V_s(s) \\ Y_{tip}(s) \end{bmatrix} = \begin{bmatrix} G_{\theta,V_a}(s) & G_{\theta,T}(s) \\ G_{V_s,V_a}(s) & G_{V_s,T}(s) \\ G_{Y,V_a}(s) & G_{Y,T}(s) \end{bmatrix} \begin{bmatrix} V_a(s) \\ T(s) \end{bmatrix} \qquad (23)$$

The subscripts p and q of each element $G_{p,q}(s) = N_{p,q}(s)/D(s)$ identify the associated output and input variable respectively. Each of the six transfer functions for this system have the common denominator $D(s)$ and are given in [11].

4.1   Rational Laplace Domain Transfer Functions

The transfer functions presented above are exact relationships (for the model used) in terms of transcendental functions of β. The desired form of the transfer functions (23) is in terms of ratios of polynomials in the Laplace operator s. For this purpose, the numerator and denominator transcendental functions can be rationalized as shown by Schmitz [10] using the Maclaurin series expansion,

$$f(\beta) = \frac{\beta^p}{p!} \frac{d^p f(0)}{d\beta^p}(1 + k_1\beta^4 + k_2\beta^8 + \ldots) \qquad (24)$$

or in equivalent product form

$$f(\beta) = \frac{\beta^p}{p!} \frac{d^p f(0)}{d\beta^p} \prod_{i=1}^{N} \left[1 - \left(\frac{\beta}{\beta_i}\right)^4\right] \qquad (25)$$

The derivative term in the leading coefficient $\frac{d^p f(0)}{d\beta^p}$ is obtained by differentiating the function $f(\beta)$ until a non-zero constant is obtained. The p is the index of that first non-zero derivative and N corresponds to the

number of modes to be retained in the rationalized representation. This procedure leads to the following rational expressions needed for this work from (23) in terms of s.

$$G_{V_s,T}(s) = -\frac{C_s \rho A}{24 I_T EI} \left(4l \left((l-x_1)^3 - (l-x_2)^3 \right) - \left((l-x_1)^4 - (l-x_2)^4 \right) \right) \prod_{i=1}^{N} \frac{((s/\gamma_{T_i})^2 + 1)}{((s/\Omega_i)^2 + 1)} \qquad (26)$$

$$G_{\theta,V_a}(s) = -\frac{C_a}{C_s} G_{V_s,T}(s) \qquad (27)$$

where the total rotary inertia I_T is defined $I_T \triangleq I_h + \frac{l^3 \rho A}{3}$.

5 IMSC

Given a linear transfer function representation of a flexible structure system in the form:

$$Y(s) = G(s)F(s) \qquad (28)$$

where:

$F(s)$ is a vector of control input forces,

$G(s)$ is a matrix of transfer functions,

$Y(s)$ is a vector of outputs,

The system output can be expressed as:

$$Y(s) = \sum_{i=1}^{n} \eta_i(s) + \delta U(s) = \sum_{i=1}^{n} \left(\frac{(B_i s + C_i) U(s)}{(s+c_i)^2 + d_i^2} \right) + \delta U(s) \qquad (29)$$

from Appendix A (58), where $\eta_i(s)$ is the Laplace domain form of the ith modal coordinate. Equation (29) implies that in the time domain the ith modal differential equation of a system with proportional damping and multiple inputs can be written as

$$\ddot{\eta}_i(t) + 2\zeta_i p_i \dot{\eta}_i(t) + p_i^2 \eta_i(t) = B_i \dot{u}(t) + C_i u(t) + f_i(t) \qquad (30)$$

where ζ_i is the ith modal damping ratio, $f_i(t)$ the ith modal independent control force, $u(t)$ the $n \times 1$ disturbance input vector, and $B_i = [B_{i1} B_{i2} \ldots B_{in}]^T$ the ith modal participation vector corresponding to the inputs derivative and C_i being likewise for the inputs. Since each element f_i of the modal control vector is designed to depend only upon the corresponding i^{th} modal displacement and velocity, n representing the number of actively controlled modes, then the closed loop system remains decoupled and the control is referred to as Independent Modal Space Control (IMSC). The IMSC method has several desirable characteristics [4], [5], [6] including simplicity in design and implementation, and relatively good robustness properties. It also has some undesirable features, namely, that the favorable robustness and closed-loop modal independence

properties cited above are lost when the self-adjointness requirement is not satisfied, and that the number of actuators required is equal to the number of controlled modes.

If the modal control law is defined as

$$f_i(t) = -g_i \eta_i(t) - h_i \dot{\eta}_i(t) \tag{31}$$

the ith modal equation of the closed loop system becomes

$$\ddot{\eta}_i(t) + (2\zeta_i p_i + h_i)\dot{\eta}_i(t) + (p_i^2 + g_i)\eta_i(t) = B_i \dot{u}(t) + C_i u(t) \tag{32}$$

This equation is much easier to solve if we assume very light damping and hence remove the $B_i \dot{u}(t)$ term. Suppose that the inputs consist of n unit impulse disturbances, (32) can be solved using Laplace transformation [12] as:

$$\eta_i(t) = \frac{c_i}{b_i} e^{-a_i t} \sin b_i t \tag{33}$$

where

$$c_i = \sum_{j=1}^{n} B_{ij} \tag{34}$$

$$a_i = \eta_i p_i + \frac{h_i}{2} \tag{35}$$

$$b_i = \sqrt{p_i^2(1 - \eta_i^2) + g_i - \eta_i p_i h_i - \frac{h_i^2}{4}} \tag{36}$$

For independent modal space control, the performance index can be expressed as a linear combination of modal performance indices as

$$J = \sum_{i=1}^{n} J_i \tag{37}$$

where each modal index J_i can be minimized independently. In the work presented here, the modal performance indices are defined as

$$J_i = \int_0^\infty \left[\dot{\zeta}_i^2 + (p_i^2 + gi)\zeta_i^2 + \frac{1}{2\alpha^2} f_i^2 \right] dt \tag{38}$$

The first two terms in the integral in (38) represent the total mechanical energy corresponding to the ith mode of the closed loop system defined in (32), and the third term in the integral is the control effort with a penalty factor α. Substituting (31) and (33) into (38) and performing the integration yields

$$J_i = \left[2 + \frac{h_i^2}{2\alpha^2} + \frac{g_i^2}{2\alpha^2(p_i^2 + g_i)} \right] \frac{c_i^2}{4\zeta_i p_i + 2h_i} \tag{39}$$

The modal index J_i can be minimized with respect to g_i and h_i by setting the corresponding first partial derivatives equal to zero. This procedure renders the following optimal gains:

$$g_i^{opt} = 0 \tag{40}$$

$$h_i^{opt} = -2\zeta_i p_i + 2\sqrt{\zeta_i^2 p_i^2 + \alpha^2} \tag{41}$$

12 - 9

Examination of (31), (32), (40) ,and (41) indicates that in the optimal design presented here the control effort is used only for increasing the damping of the closed loop system, and no effort is expended to change the natural frequencies of the system. The total performance index for the optimally controlled structure, J_a^{opt}, can be obtained by substituting (39), (40) and (41) into (37), i.e.,

$$J_a^{opt} = \frac{1}{\alpha^2} \sum_{i=1}^{n} c_i^2 \left(-\zeta_i p_i + \sqrt{\zeta_i^2 p_i^2 + \alpha^2} \right) \tag{42}$$

This partially optimized index will be further minimized using a designed-in viscoelastic passive damping treatment.

5.1   Implementation of IMSC

For implementation we want to take the developed equations from above and represent them in a State-Space formulation. From (29) it is possible to convert the transfer function to a partial fraction expansion form:

$$Y = \sum_{j=1}^{m} \left(\sum_{i=1}^{n} \frac{B_{ij} s + C_{ij}}{(s + c_i)^2 + d_i^2} + \delta_j \right) F_j = \sum_{i=1}^{n} \eta_i + \sum_{j=1}^{m} \delta_j F_j \tag{43}$$

where n is the number of modes and m is the number of inputs and once again F_i is a vector of input control forces. From here we can deduce that:

$$\eta_i = \sum_{j=1}^{m} \frac{B_{ij} s + C_{ij}}{(s_i + c_i)^2 + d_i^2} F_j \tag{44}$$

In this particular experiment there are two inputs, so to simplify this development, let $j = 1, 2$ and

$$\eta_i = \frac{(B_{i1} s + C_{i1}) F_1 + (B_{i2} s + C_{i2}) F_2}{(s + c_i)^2 + d_i^2} \tag{45}$$

using Laplace Transform rules. this can be converted directly to the second order, time domain modally decoupled form:

$$\ddot{\eta}_i + 2c_i \dot{\eta}_i + (c_i^2 + d_i^2) \eta_i = B_{i1} F_i + B_{i2} F_2 + C_{i1} \dot{F}_1 + C_{i2} \dot{F}_2 \tag{46}$$

where c_i comes from (34) and contains the ith modal damping ratio, B and C come from the pfe, and F and $\dot{F}$ being the input force and its time derivative, respectively. From (46), the independent modal forces are:

$$f_i = B_{i1} \dot{F}_1 + B_{i2} \dot{F}_2 + C_{i1} F_1 + C_{i2} F_2 \tag{47}$$

Solving (46) for $\ddot{\eta}_i$ and integrating twice:

$$\eta_i = -2c_i \int_0^\infty \eta_i dt + B_{i1} \int_0^\infty F_1 dt + B_{i2} \int_0^\infty F_2 dt - (c_i^2 + d_i^2) \int_0^\infty \int_0^\infty \eta_i dt dt + \tag{48}$$
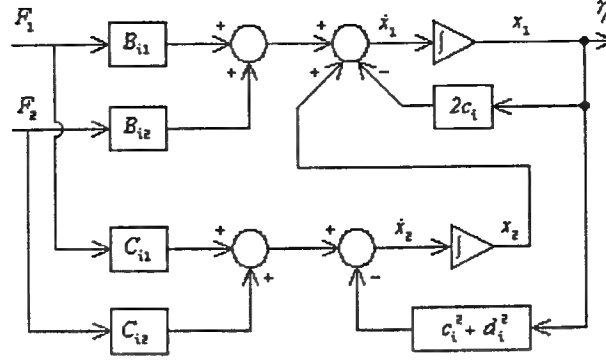$$C_{i1} \int_0^\infty \int_0^\infty F_1 dt dt + C_{i2} \int_0^\infty \int_0^\infty F_2 dt dt$$

12 - 10

Figure 1: Block Diagram Showing Relationship of Input Control Force to Modal Displacement

Now if we let

$$\dot{x}_1 = -2c_i\eta_i + B_{i1}F_1 + B_{i2}F_2 + x_2$$

$$\dot{x}_2 = -(c_i^2 + d_i^2)\eta_i + C_{i1}F_1 + C_{i2}F_2$$

$$x_1 = \eta_i$$

This is illustrated in block diagram form in figure (1).

Note that $x_2 \neq \dot{\eta}_i$, but $\dot{x}_1$ can be formed in the observer. It is easy to see the following State-Space representation:

$$A_{pl} = \begin{bmatrix} -2c_i & 1 \\ -(c_i^2 + d_i^2) & 0 \end{bmatrix} \quad B_{pl} = \begin{bmatrix} B_{i1} & B_{i2} \\ C_{i1} & C_{i2} \end{bmatrix}$$

$$C_{pl} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad D_{pl} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

where A_{pl}, B_{pl}, C_{pl}, and D_{pl} were used for the state-space form, to avoid confusion with the same symbols used previously. Also note here that A_{pl} should be represented in block-diagonal form with dimension $2i \times 2i$. B_{pl}, C_{pl}, and D_{pl} are in column form and have dimensions $2i \times 2$, $1 \times 2i$, and 1×2 respectively. Next an observer is needed to get $\dot{\eta}_i$. Therefore we want the output of the observer to be

$$y_i = \begin{bmatrix} \eta_i \\ \dot{\eta}_i \end{bmatrix} \tag{49}$$

So refering back to figure (1), the C_{ob}, and D_{ob} matrices for the observer are

$$C_{ob} = \begin{bmatrix} 1 & 0 \\ -2c_i & 1 \end{bmatrix} \quad D_{ob} = \begin{bmatrix} 0 & 0 \\ B_{i1} & B_{i2} \end{bmatrix}$$

Now we should develop the rest of the observer. From [13]

$$\dot{\hat{x}} = (A - KC)\hat{x} + Bu + Ky \tag{50}$$

12 - 11

where $\hat{x}$ is an estimate of x, the state variables A, B, and C are the plant's A, B, C matrices, K is a matrix of Kalman gains, and u and y are the plants input and output. We already know that the output and input vectors are

$$y = C\hat{x} + Du \quad u = -G\hat{x} \tag{51}$$

where G contains the optimal gains from (40) and (41) and D is the plant D matrix. Therefore (50) becomes

$$\dot{\hat{x}} = (A - K(C + DG) - BG)\hat{x} + Ky \tag{52}$$

substituting in the y from (51) and simplifying

$$\dot{\hat{x}} = (A - KC - BG)\hat{x} + KCx \tag{53}$$

Now representing the observer in State-Space form

$$A_{ob} = [(A_{pl} - KC_{pl} - B_{pl}G)] \quad B_{ob} = [KC_{pl}]$$

So thus far we have the plant model, which has an output of modal displacements and an observer which has an output of modal displacements and modal velocities, η_i and $\dot{\eta}_i$. From here we have to get the control force. For implementation reasons it will be much easier to assume small damping and drop the $B_{ij}\dot{F}_j$ terms in (47). Therefore

$$f_i = C_{i1}F_1 + C_{i2}F_2 \tag{54}$$

Now we have to keep in mind that f_i is the modal input to mode i and not the input to any one particular actuator. The individual actuator inputs are the F_j's. To make it clearer what has to be done, (54) rewritten for two modes is

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \tag{55}$$

The matrix containing the C_{ij} terms will be denoted as C. By multiplying (55) by C^{-1}

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}^{-1} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \tag{56}$$

We also have to remember the feed-forward term from (43), δ.

6 Experimental Results

The experiment consists of a 2 m long uniform aluminium beam of rectangular cross section (76.2 mm × 6.35 mm), hanging vertically from a very low friction hinge. The hinge is in a knife edge arrangement and is situated in such a way that the root of the flexible beam is coincident with the hinge axis. This also allows PZT actuator/sensor pairs to be located with one edge coincident with the hinge axis. The hinge arrangement has effectively zero inertia, however, a hub inertia has been added in order to increase

pole-zero separation. Because the angular displacements of the hinged joint are very small, it was possible to get an adequate measurement of hub angle by measuring translational displacement of a point on the arm extending from the hinge. A Kaman KD-2300-10CU non-contact displacement measuring system was used for this measurement. This sensor has a rated 3dB frequency response range of static to 50 kHz., a measurement range of 25.4 mm, and 0.254 mm midrange resolution. The piezoelectric ceramic elements are Vernitron PZT5A patches mounted in pairs on opposite faces of the beam. A 63.5 mm long by 38.1 mm wide piezoelectric material patch is attached to each side of the root of the beam so $x_1 = 0$, $x_2 = 63.5$ mm, and $w = 38.1$ mm. A second pair is located roughly midway down the beam at $x_1 = 1.048$ m, $x_2 = 1.111$ m. The two pairs of patches were tied together to effectively make one actuator at the root and one in the middle of the beam.

A signal generator was used as input to the patches for excitation. When the beam's motion reached what appeared to be steady-state the generator was turned off and the controller, on. The controller was implemented on a Sun SPARCstation equiped with a Solflower input/output device, where the Kaman sensor signal was the input and the two control voltages for the patches was the output. A C-language program which read the discretized A, B, C, D matrices from a file was used and operated at 0.6 kHz. The two signals from the Solflower were each sent to an amplifier and then to their respective piezoelectic patch-pairs. A Tektronix model 2630 Fourier Analyzer was used for data acquisition. The parameters for the experimental system are tabulated in Table 1. Table 2 contains the parameters of the piezoelectric material.

Hub inertia, I_h	0.0348 N-m-s^2	Beam length, l	2.0 m
Volumetric mass density, ρ	2712.6 Kg /m^3	Cross sectional area, A	483.87 $\times 10^{-6}$ m^2
Young's Modulus, E	69.0×10^9 N/m^2	Area moment of inertia, I	1.63×10^{-9} m^4

Table 1: Experimental Beam Parameters

Charge Constant, d_{31}	-171×10^{-12} m/v	Voltage Constant, g_{31}	-11.4×10^{-3} Vm/N
Coupling Coefficient, k_{31}	-0.340	Capacitance, C	68.35 μF/m^2
Surface area, $w \times (x_2 - x_1)$	0.0024 m^2	Thickness t_a	3.05×10^{-4} m
Young's Modulus, E_a	69.0×10^9 N/m^2		

Table 2: PZT5A Parameters

On this beam there are two actuators so two modes can be simultaneously controlled using IMSC. Due to a lack of voltage amplification the controller was implemented for one mode at a time. The amplifiers used had a maximum power output of ± 35 volts. It isn't until about ± 25 volts that the these particular patches seem to respond with sufficient control authority. As it turned out for a particular α, the voltage for patch-pair one was an order of magnitude higher than that for patch two. So for example with the beam
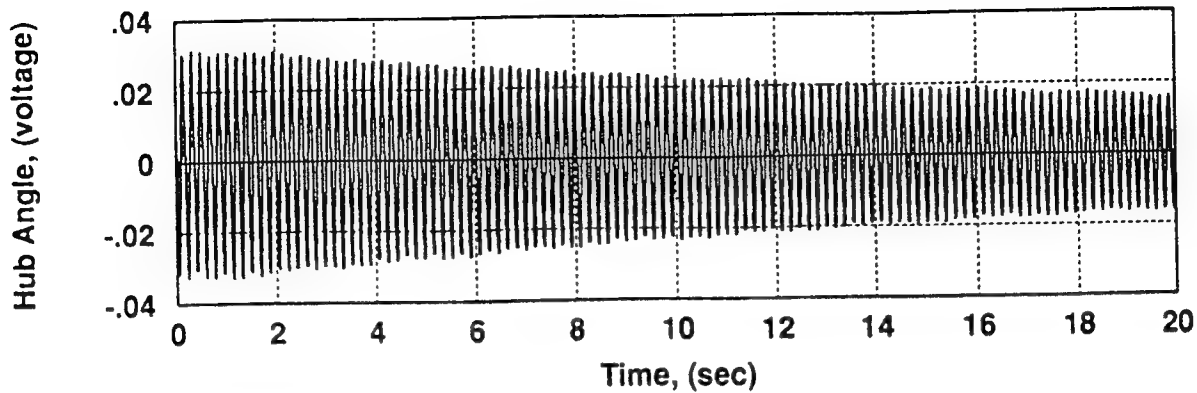
Figure 2: Mode 1, Free Time Response, (experimental)

vibrating in the first mode, and α adjusted so the voltage for patch-pair one was just saturating its amplifier at ±35 volts, the voltage to patch-pair two would be near ±3.5 volts. Therefore the second patch-pair would not be activated. This is acceptable for controlling the second mode. Yet the first mode is best controlled with the middle patch. So it was decided to implement the controller for each mode separately. Since mode one is best controlled from the second patch-pair, α was adjusted so that when the controller was first turned on, the amplifier for patch-pair two just barely saturated. This of course makes for a control voltage for the first patch-pair that is far to high for the available amplifier. Therefore the connection for the first patch-pair was disconnected. In this configuration, significant damping was achieved as seen in figure (4). The simulation for controlling mode one is shown in figure (5) and correlates well with (4). As previously stated mode two is best controlled using the first patch-pair. Therefore mode two was controlled using the same α with the patch-pair at the root connected. The particular α used to control mode one made the amplifier that supplied the first patch-pair, just saturate. This is exactly what is needed to get the most control authority from the patches. The free time response plot for the beam with mode two controlled is shown in figure (8) and the simulation is shown in figure (9). Comparison of the two show a very nice correlation between the modeled and actual implementation. Future implementations will include optimal passive damping in conjuction with this optimal active control.

7 Conclusion

An optimal method for actively controlling a flexible slewing piezelectric laminate beam has been presented. The method uses the partial fraction expansion of exact open loop transfer functions of the beam and is implemented using Independant Modal Space Control (IMSC). Implementation of the controller was
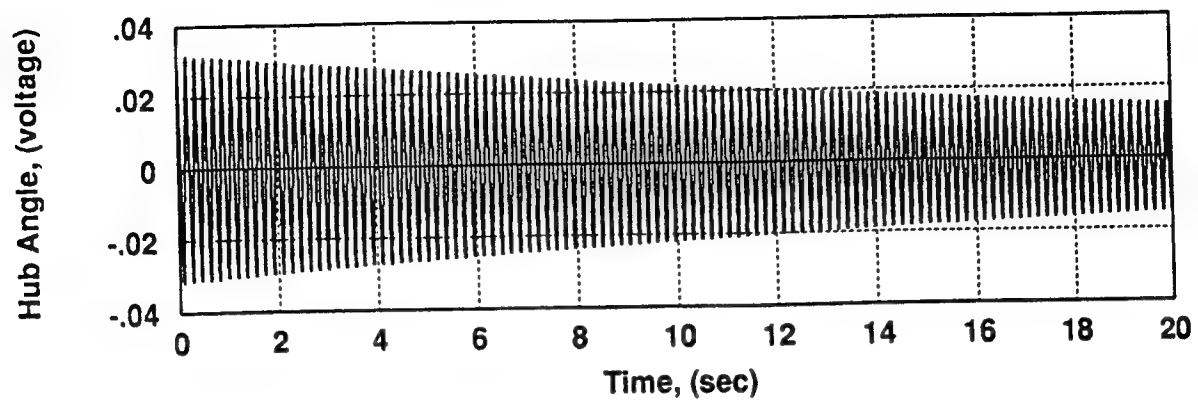
Figure 3: Mode 1, Free Time Response, (simulated)
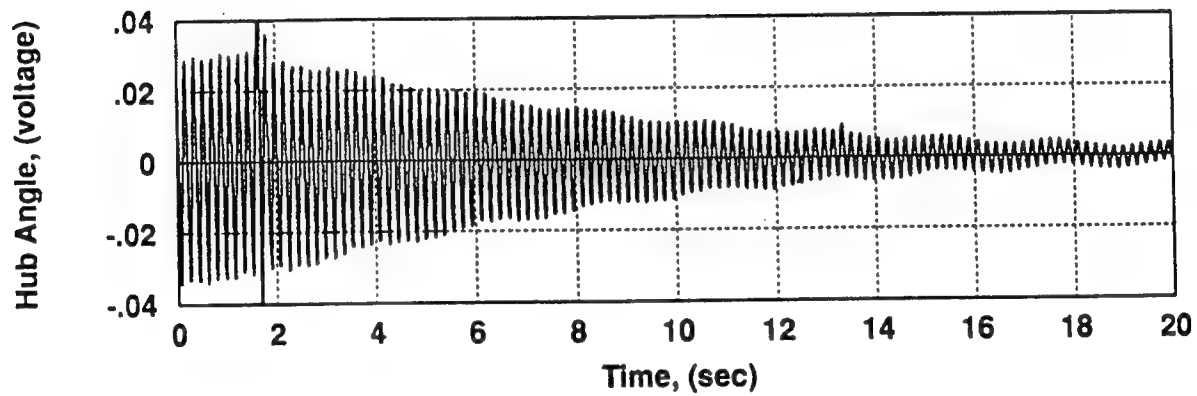
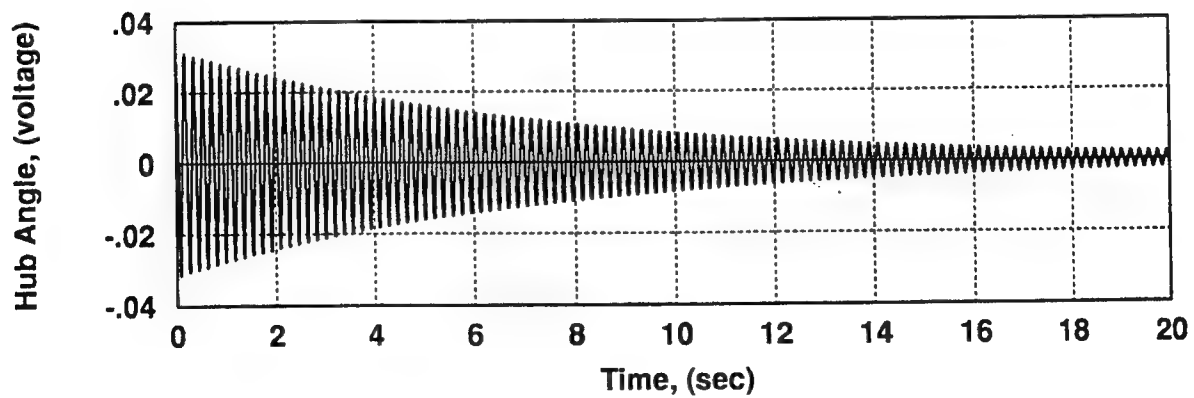Figure 4: Mode 1. Controller On, Free Time Response, (experimental)

Figure 5: Mode 1, Controller On, Free Time Response, (simulated)
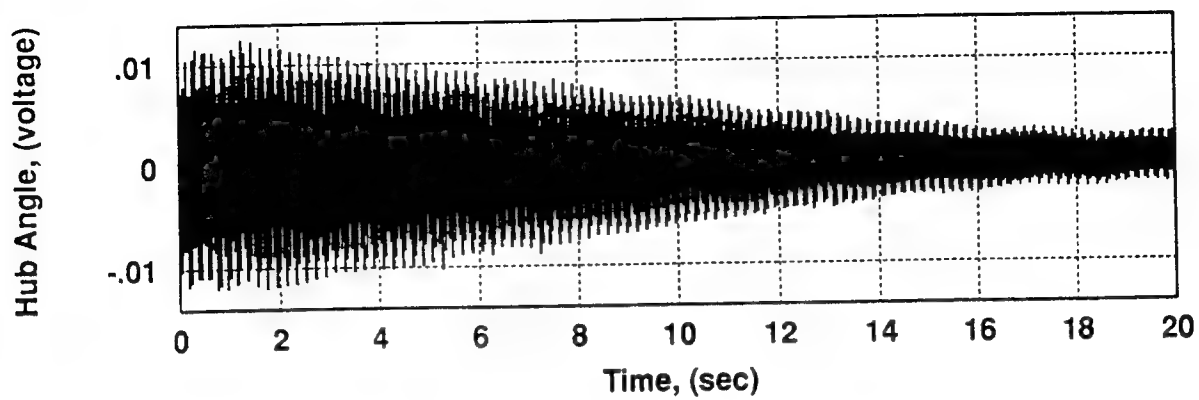
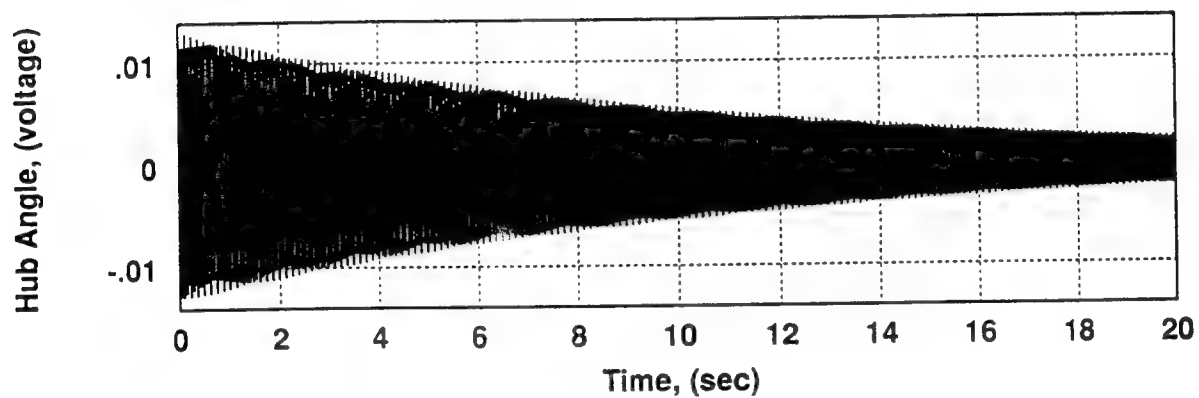Figure 6: Mode 2, Free Time Response, (experimental)
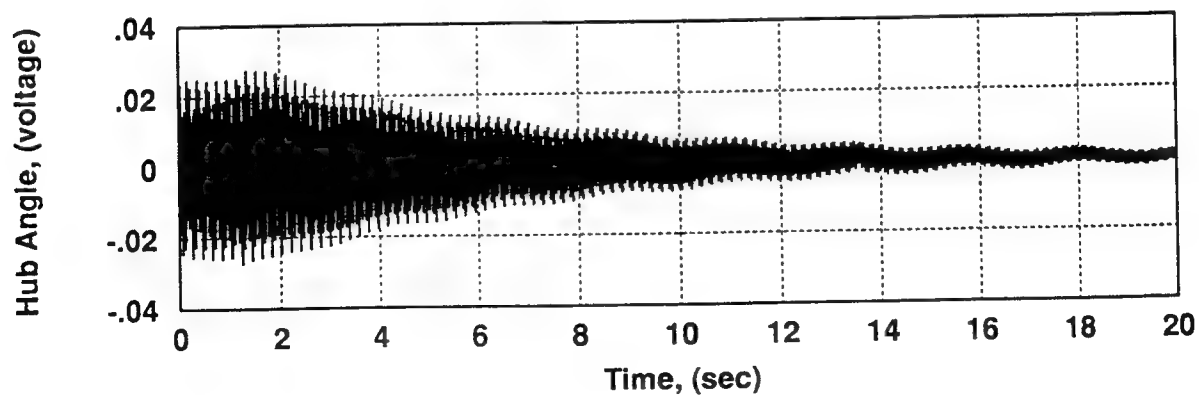

Figure 7: Mode 2, Free Time Response, (simulated)

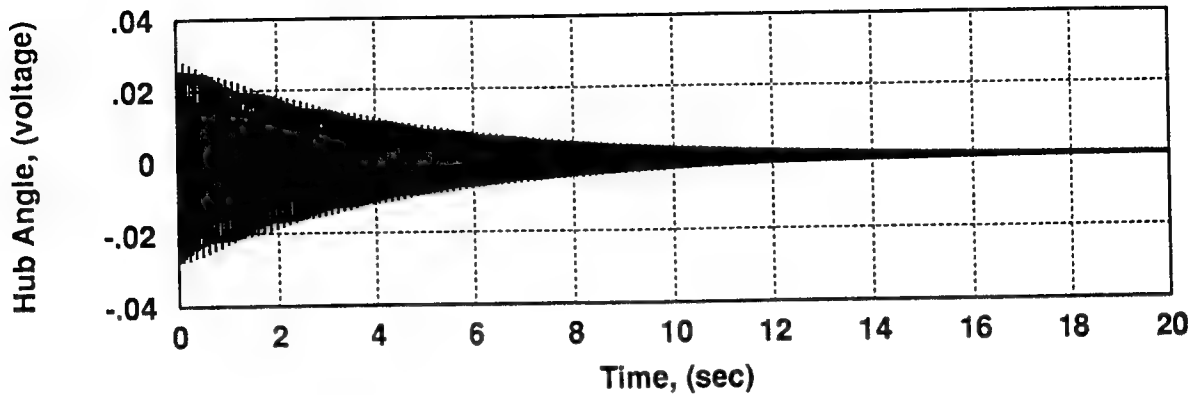Figure 8: Mode 2, Controller On, Free Time Response, (experimental)

Figure 9: Mode 2, Controller On, Free Time Response, (simulated)

conducted at the US Air Force Academy, Frank J. Seiler Research Laboratory. The results show significant vibration reduction for the very flexible beam presented here. This work leads directly to the implementation of an optimal active and passive control design which uses a constrained viscoelastic damping layer.

8   Acknowledgement

A   Partial Fraction Expansion

The partial fraction expansion of a transfer function with a product of m second order terms $s^2 + 2\zeta_i z_i s + z_i^2$ as the numerator, and n second order terms of the form $s^2 + 2\zeta_i p_i s + p_i^2$ as its denominator can be expressed as:

$$G(s) \quad = \quad \frac{\displaystyle\prod_{i=1}^{m}(s + a_i)^2 + b_i^2}{\displaystyle\prod_{i=1}^{n}(s + c_i)^2 + d_i^2}$$

$$= \frac{\prod_{i=1}^{m}(s^2 + 2\zeta_i z_i s + z_i^2)}{\prod_{i=1}^{n}(s^2 + 2\zeta_i p_i s + p_i^2)}$$

therefore

$$a_i = \zeta_i z_i \quad b_i = z_i \sqrt{1 - \zeta_i^2}$$

$$c_i = \zeta_i p_i \quad d_i = p_i \sqrt{1 - \zeta_i^2} \tag{57}$$

Next we can express the function in the general partial fraction expansion form:

$$G(s) = \sum_{i=1}^{n}\left(\frac{A_i}{s + c_i + jd_i} + \frac{A_i^*}{s + c_i - jd_i}\right) + \delta = \sum_{i=1}^{n}\left(\frac{B_i s + C_i}{(s + c_i)^2 + d_i^2}\right) + \delta$$

where

$$\delta = \begin{cases} 1 & n = m \\ 0 & n > m \end{cases}$$

For any r^{th} term the first order residue A_r is:

$$A_r = (s + c_r + jd_r)\left.\frac{\prod_{i=1}^{m}(s + a_i)^2 + b_i^2)}{\prod_{i=1}^{n}(s + c_i)^2 + d_i^2)}\right|_{s=-c_r-jd_r}$$

$$= \frac{1}{(-2jd_r)}\left.\frac{\prod_{i=1}^{m}(s + a_i)^2 + b_i^2}{\prod_{i=1}^{r-1}(s + c_i)^2 + d_i^2 \prod_{i=r+1}^{n}(s + c_i)^2 + d_i^2}\right|_{s=-c_r-jd_r}$$

$$= \frac{j}{(2d_r)}\frac{\prod_{i=1}^{m}((-c_r + a_i) - jd_r)^2 + b_i^2}{\prod_{i=1}^{r-1}((-c_r + c_i) - jd_r)^2 + d_i^2 \prod_{i=r+1}^{n}((-c_r + c_i) - jd_r)^2 + d_i^2}$$

$$= \frac{j}{(2d_r)}\frac{\prod_{i=1}^{m}(a_i - c_r)^2 + b_i^2 - d_r^2 - 2jd_r(a_i - c_r)}{\prod_{i=1}^{r-1}(c_i - c_r)^2 + d_i^2 - d_r^2 - 2jd_r(c_i - c_r) \prod_{i=r+1}^{n}(c_i - c_r)^2 + d_i^2 - d_r^2 - 2jd_r(c_i - c_r)} \tag{58}$$

It is not necessary to get A_r^* explicitly. From (58):

$$\frac{B_r s + C_r}{(s + c_r)^2 + d_r^2} = \frac{A_r}{s + c_r + jd_r} + \frac{A_r^*}{s + c_r - jd_r}$$

$$= \frac{A_r(s + c_r - jd_r) + A_r^*(s + c_r + jd_r)}{(s + c_r)^2 + d_r^2}$$

$$= \frac{(s + c_r)(A_r + A_r^*) + jd_r(A_r^* - A_r)}{(s + c_r)^2 + d_r^2}$$

$$= \frac{2(s + c_r)Re(A_r) + 2d_r Im(Ar)}{(s + c_r)^2 + d_r^2}$$

Note that the Re() term means real part and Im() means imaginary part. From here we see:

$$B_r = 2Re(A_r)$$

$$C_r = 2(c_r Re(A_r) + d_r Im(A_r))$$

These two formulas seem a little clumsey with the Re and Im comments, but are very straight forward to implement in a computer code.

References

[1] M.J. Balas, "Feedback Control of Flexible Systems", IEEE Transactions on Automatic Control, VOL.AC-23, No.4, Aug. 1978, pp. 673-679.

[2] J. Fanson, G. Blackwood, and C.-C. Chu, "Experimental Evaluation of Active-Member Control of Precision Structures", Presented at the NASA/DOD CSI Conference, Jan.29 - Feb.2, 1989.

[3] T.E. Alberts, "Augmenting the Control of a Flexible Manipulator with Passive Mechanical Damping", Ph.D. Dissertation, Georgia Institute of Technology, Sept. 1986.

[4] L. Meirovitch and L.M. Silverberg, "Globally Optimal Control of Self-Adjoint Distributed Systems", Optimal Control Applications and Methods, Vol. 4, 1983, pp. 365-386.

[5] L.M. Silverberg., "Uniform Damping Control of Spacecraft", Journal of Guidance Control, and Dynamics, Vol. 9, 1986, pp. 221-226.

[6] H. Baruh and L. Meirovitch, "Implementation of the IMSC Method by Means of a Varying Number of Actuators", Proceedings of the Fourth VPI&SU/AIAA Symposium on Dynamics and Control of Flexible Structures, 1983, pp. 605-616.

[7] A. Baz and S. Poh, "Performance of an Active Control System with Piezoelectric Actuators," J of Sound and Vibrations, Vol 126, No 2, PP 327–343, 1988.

[8] T. Bailey and J. E. Hubbard, "Distributed Piezoelectric-Polymer Active Vibration Control of a Cantilever Beam," J. Guidance, Control and Dynamics, Vol 8, No 5, pp 605–611, 1985.

[9] A. Papoulis, *The Fourier Integral and its Applications*, McGraw-Hill, 1962.

[10] E. Schmitz, "Experiments on the End-Point Position Control of A Very Flexible One-Link Manipulator," Ph.D. Dissertation, Stanford University, Stanford CA., 1985.

[11] T.E. Alberts, and T.V. DuBois, "Multivariable Transfer Functions and Optimal Passive Damping for a Slewing Piezoelectric Laminate Beam," Final Report for the AFOSR Summer Research Program, FJSRL, Jul. 31, 1992. In review for publication in the AMSE *Journal of Dynamic Systems, Measurement, and Control*, also presented in IFAC 1993, Sydney, Australia.

[12] T.E. Alberts and Joseph V. Harrell, "Optimal Passive/Active Control Design Synergism for Flexible Structures," Submitted to the AIAA Journal of Guidance, Control and Dynamics, July 12, 1991.

[13] B. Friedland. *Control System Design. An Introduction to State-Space Methods*, McGraw-Hill, 1986.

The Rational Design of Novel Semi-Organic Salts for Use as Nonlinear Optical Materials

Joan Fuller
Graduate Student
Department of Chemistry

University of Alabama
Chemistry Department
Tuscaloosa, Al 35487

The Rational Design of Novel Semi-Organic Salts for Use as Nonlinear Optical Materials

Joan Fuller
Graduate Student
Department of Chemistry
University of Alabama

Abstract

The primary focus of this project was the rational design of nonlinear optical materials based on the concept of crystal engineering. It was proposed to use hydrogen bonding and π-stacking effects to design NLO optimized organic/inorganic salts. The salt series was based around the imidazolium cation with anions varying in size and hydrogen bonding ability. The salts were screened for second harmonic generation using the powder technique developed by Kurtz and Perry. In addition, the salts were characterized by IR, NMR, Raman, melting point, and mass spectroscopy data.

Introduction to Nonlinear Optics

Joan Fuller

Nonlinear optics is the study of the interaction of intense laser light with matter and how this interaction modifies the optical properties of the material.[1] Nonlinear optical materials are divided into two classes; second-order and third-order materials. Second-order materials have changes associated with changes in the first nonlinear term of the polarization equation. They are characterized by the molecular hyperpolarizability term, β. Third-order NLO effects are characterized by the second hyperpolarizability, γ. At this time, all commercial applications use second-order nonlinear optical effects; no third-order devices are being produced. In order to design materials that maximize the second-order NLO response it is necessary to understand the molecular and macroscopic polarization of the material. Equation 1 expresses the molecular polarization of a material in an applied electric field.[2]

$$P = \mu + \alpha \cdot E + \beta \cdot E \cdot E + \gamma \cdot E \cdot E \cdot E + \dots$$

Equation 1

μ = permanent dipole moment

α = linear polarizability

β & γ = first and second hyperpolarizabilities

E = applied electric field

The individual molecular responses interact to give the bulk response, also called the macroscopic polarization of the material, expressed by Equation 2.[2]

$$P = P_0 + \chi^1 \cdot E + \chi^2 \cdot E \cdot E + \chi^3 \cdot E \cdot E \cdot E + ...$$

Equation 2

P_0 = spontaneous polarization of the material

χ^1 = linear susceptibility

χ^2 & χ^3 = second and third order susceptibilities

NLO Effects

Several nonlinear optical effects are known but perhaps the best known is second-harmonic generation (SHG). Originally observed in single crystal quartz, SHG was the first NLO interaction to be described.[3] In an SHG measurement a laser beam of frequency ω is focused on a sample, and coherent light at twice the initial frequency 2ω is detected. Because SHG measurements can be made on single crystals, poled polymers, films, or powders, it is a common screening technique used to determine NLO activity. The molecular hyperpolarizability term responsible for SHG effects is β. In general, the higher the β value, the larger the NLO effect of the material.[2]

Two other important second-order nonlinear effects are sum-frequency generation (SFG) and difference-frequency generation (DFG). In SFG, two input waves, ω_1 and ω_2, are added to generate a new wave of frequency ω_3 ($\omega_1 + \omega_2 = \omega_3$). In DFG, two input waves, ω_1 and ω_2, are subtracted to generate a new amplified wave of frequency ω_3 ($\omega_1 - \omega_2 = \omega_3$); therefore, DFG is also referred to as optical parametric amplification.[4]

Second Order NLO Materials

Compounds that exhibit nonlinear optical properties are typically divided into three categories:

* inorganic   * organic   * semi-organic

Inorganic

Inorganic compounds are the only commercially available NLO compounds, in part because they were the first class of materials to be studied, but also because they have the desired physical properties needed for useful materials. However, there are limitations to this class of materials. They have small χ^2 values, and there are few opportunities to alter the properties beyond what can be achieved through a restricted number of substitutions. Furthermore, many of the inorganic crystals must be grown from high temperature melts which leads to growth impurities and the introduction of crystal strain due to the cooling process. Problems such as these, result in sample to sample variations that make it difficult to design operative systems.[5] Table 1 lists examples of inorganic NLO systems that are currently in use and their relative merit as materials.[6]

Table 1: Inorganic NLO Systems [6, 7]

Compound	FOM[a]
b - BaB_2O_4	26
KH_2PO_4 (KDP)	1
$KTiOPO_4$ (KTP)	215
Urea	6

a. FOM: Figure of merit, a measure of the operative efficacy relative to KDP of the NLO material in a systems application.

Organic

Organic compounds offer many advantages over inorganic compounds. The design flexibility of organic chemistry provides the opportunity to modify the response and spectral properties of the material. In addition, there are examples of organic systems with frequency conversion responses orders of magnitude larger than the best inorganic systems. However, organic systems do have their problems. First, it is difficult to grow high quality crystals. Second, the interactions between molecules and between the crystal planes is usually relatively weak, which causes the majority of these materials to lack the physical properties needed for NLO materials.[8] Finally, organic compounds crystallize in a centrosymmetric space group 75% of the time and, therefore, are useless as NLO materials unless poling experiments are performed.

The largest single class of second-order organics consists of donor-acceptor substituted aromatics. The donors can be amino or dimethyl amino groups, oxazole, dithiafulvene, and dihydropyrazole. The acceptors can be nitro, pyridinium ions, fulvenes, dicyano or tricyanovinyl groups, sulfones, enones, or imines. Aromatics are used to establish highly polarized charge distributions and can include, benzenes, stilbenes, diarylacetylenes or biaryls.[9] Table 2 lists some promising donor-acceptor systems along with their molecular hyperpolarizability values. It is obvious that these donor-acceptor materials should have high bulk SHG values if they can be made to crystallize in acentric space groups.

Table 2: Donor-Acceptor Aromatic Systems [10]

Donor	Aromatic	Acceptor	β x 10^{-30} esu
NH_2	Benzene	CN	3.1
NMe_2	Benzene	CN	5.0
NMe_2	Benzene	NO	12
NMe_2	Benzene	NO_2	12
NMe_2	Stilbene	NO_2	73
NMe_2	Stilbene	CN	36
OMe	Stilbene	NO_2	34
NH_2	Stilbene	NO_2	40

* Urea has a β value of ~0.5 x 10^{-30}esu.

Semi-organic

Semi-organics, or mixed salts, are the newest class of materials studied for NLO applications. They combine the best of the organic and inorganic systems. Perhaps their greatest advantage, is their straightforward synthesis that allows them to be rapidly screened for SHG activity. Two general categories of semi-organics have been defined: Type I: Salts containing an NLO active organic ion and an inorganic counterion, and Type II: Coordination complex salts containing a neutral NLO organic ligand coordinated to a metal ion. Tables 3 & 4 lists examples of Type I and Type II materials.

Table 3: Type I Semi-organic Systems [11]

Cation	Anion	SHG Value vs Urea
$4\text{-}CH_3OC_6H_4CH=CHC_5H_4NCH_3$	CF_3SO_3	50
$4\text{-}CH_3OC_6H_4CH=CHC_5H_4NCH_3$	BF_4	0
$4\text{-}CH_3OC_6H_4CH=CHC_5H_4NCH_3$	$CH_3C_6H_4SO_3$	120
$4\text{-}CH_3OC_6H_4CH=CHC_5H_4NCH_3$	Cl	60

Table 4: Type II Semi-organic Systems [12]

Compound	SHG Value vs KDP
$Cd(thiourea)_2Cl_2$	2.75
$Zn(thiourea)_3SO_4$	3.6
$Pb(diethyldithiocarbamate)_2$	12

In these systems the organic molecule provides the source of nonlinear polarization, and offers the flexibility needed to design working NLO systems. The inorganic component provides the strong three dimensional binding network that gives the mechanical stability. Another advantage of these systems is the ability to crystallize them at ambient temperature, thus, creating less strain in the crystal and making commercial applications less expensive.[12]

Methodology

A series of Type I semi-organic NLO materials listed in Table 5 were prepared and characterized. The salt was composed of the 1,3-disubstituted imidazolium cation, illustrated in Figure 1, and specific anionic groups that have been shown to alter the NLO properties of materials. These salts often stack in flat, one-dimensional layers which are promoted by hydrogen bond networks. Because of the inherent lack of symmetry that can be built into the cation, it is possible to promote crystallization into acentric space groups. The imidazolium salts prepared in

this study are soluble, stable, cheap, and are easily grown into large single crystals. The research was conducted in two phases.
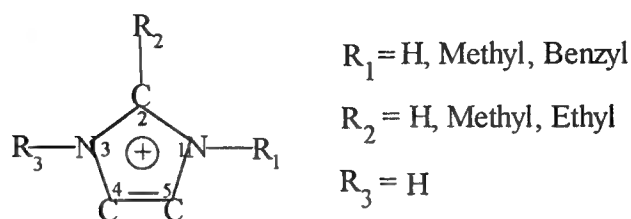

$R_1 = H$, Methyl, Benzyl

$R_2 = H$, Methyl, Ethyl

$R_3 = H$

Figure 1

Phase I: Synthesis of stable imidazolium salts as prototypes for SHG active materials

Listed in Table 5 are the salts which have been prepared. The synthesis of 1-benzylimidazolium tetrafluoroborate is given as an example of the method used to prepare the salts. In 50ml of water 2.005 g (0.0127 moles) of 1-benzylimidazole was dissolved with slight heating. While stirring this solution 2.891 g (0.0158 moles) of 48% HBF_4 was added dropwise. The solution was then rotoevaporated down to dryness. The salt was dried under vacuum at 60°-90° C for 24 hours. The reaction occurred in high yield and the product was confirmed with mass spectroscopy. Recrystallization was attempted from appropriate solvents.

Phase II: SHG screening and structural characterization

The apparatus for measuring the SHG activity of powders has been previously described by Kurtz and Perry.[13] Preliminary powder SHG measurements were determined by using 1064-nm (SH frequency at 532-nm) fundamental radiation. All measurements were referenced to urea which was measured at the beginning of each experiment. Because of problems with particle size, absorption, and fluorescense it was not possible to obtain quantitative information on the value of these salts as nonlinear optical materials. However, by noting the presence of green light (532-nm) at the sample, it was possible to determine a qualitative SHG result. Table 6 lists the results from these initial qualitative measurements. It is hoped in the future that more quantitative measurements can be made. This is not presently possible with the equipment that is available at the host laboratory.

Table 5: Melting Point ($^\circ$C) and Physical State

	NO_3^-	SO_4^{-2}	BF_4^-	Br^-	$CF_3SO_3^-$
1-Methyl-imidazolium	69-73 white solid	60-62 white solid	hydroscopic white solid	clear oil	82-85 white solid
1-Benzyl-imidazolium	94-95 white solid	clear oil	42-50 yellow solid	white solid	yellow oil
2-Methyl-imidazolium	158-159 white solid	176-178 white solid	60-62 hydroscopic white solid	120-122 white solid	yellow oil
2-Ethyl-imidazolium	93-95 white solid	clear oil	clear oil	122-123 white solid	yellow oil

Table 6: Second Harmonic Generation Results

	NO_3^-	SO_4^{-2}	BF_4^-	Br^-	$CF_3SO_3^-$
1-Methyl-imidazolium	NO	NO	NO	____	NO
1-Benzyl-imidazolium	NO	____	YES 3 x Urea	YES*	YES
2-Methyl-imidazolium	NO	NO *	NO	NO*	YES
2-Ethyl-imidazolium	NO	____	____	YES* 1 x Urea	YES

* sample fluoresces at 514-nm

Melting point data was determined from a Perkin Elmer differential scanning calorimeter. In the cases where a visual observation was necessary to check for decomposition a standard capillary melting point apparatus was used. Infrared data was obtained on a Nicolet 510P FT-IR spectrometer. Spectra for the crystalline salts were made by using KBr pellets. Salts which were oils were measured neat. Raman spectra was recorded at the University of Alabama on a Yobin U1000 Laser Raman spectrometer. Measurements were made using the 514-nm wavelength. It was believed that some of the salts were undergoing fluorescence as they generated the SHG frequency, 532-nm. If this was the case then it is possible that they would also fluoresce when exposed to the 514-nm wavelength of the Raman measurement.

Mass spectroscopy was used to evaluate the purity of the salts. These measurements were performed on a Hewlett Packard 5985B mass spectrometer by Mr. Jack Greathouse. The salts were analyzed via a direct insertion probe over a temperature range of 25°-250° C. Nuclear magnetic resonance data was obtained on a Varian Gemini 300 spectrometer. Both ^{1}H and ^{13}C were recorded for the salts. The solvent used was dimethylsulfoxide with tetramethylsilane as the internal reference.

Table 7a: IR, Raman, and NMR Data

	NO_3^-	SO_4^{-2}	BF_4^-	Br^-	$CF_3SO_3^-$
1-Methyl-imidazolium	IR: 3150, 3500, 1790, 1400, 1040, 819 cm^{-1}	IR: 3150, 3450, 1590, 1554, 1015, 400 cm^{-1}	IR: 3450, 3120, 1120, 1063, 400 cm^{-1}	IR: not available	IR: 3157, 1253, 1032, 837, 758, 640, 518 cm^{-1}
	Raman: 1039, 1322, 716 cm^{-1}	Raman: 1105, 611, 983, 450 cm^{-1}	Raman: 1070, 777, 533, 360 cm^{-1}	Raman: not available	Raman: not available
	NMR: ^{1}H δ;; 8.9(s), 7.6(d), 4.1(s) ^{13}C δ;; 136, 123, 119.5, 35	NMR: ^{1}H δ; 8.35(s), 7.4(s), 7.3(s), 3.8(t) ^{13}C δ; 137, 124.2, 122.2, 34.4	NMR: ^{1}H δ; 9.0(s), 7.7(d), 3.8(m) ^{13}C δ; 136.2, 123.5, 120.1, 35.7	NMR: ^{1}H δ; 9.3(s), 7.8(s), 7.7(s), 4.0(s) ^{13}C δ; 135.8, 123.5, 119.7, 36.63	NMR: ^{1}H δ; 9.03(s), 7.7(m), 3.87(s) ^{13}C δ; 136.2, 123.5, 120.1, 35.8

Table 7b: IR, Raman, and NMR Data

	NO_3^-	SO_4^{-2}	BF_4^-	Br^-	$CF_3SO_3^-$
1-Benzyl-imidazolium	IR: 3200, 1049, 1039, 970, 820 cm^{-1}	IR: 3170, 1190, 810, 420 cm^{-1}	IR: 3410, 3100, 1820, 1060, 1015, 900 cm^{-1}	IR: 2850, 2600, 910, 650 cm^{-1}	IR: not available
	Raman: 1354, 1209, 1039, 826 cm^{-1}	Raman: 1105, 983, 611, 450 cm^{-1}	Raman: 1070, 777, 533, 360 cm^{-1}	Raman: 1434, 997, 803, 608, 327 cm^{-1}	Raman: not available
	NMR: ^{1}H δ; 9.3(s), 7.8(s), 7.7(s), 7.4(s) 5.4(s) ^{13}C δ; 135, 128.5, 122, 120, 51.5	NMR: ^{1}H δ; 8.7(s), 7.5(s), 7.4(s), 5.4(s) ^{13}C δ; 136.6, 128.6, 124, 121.4, 51.1	NMR: ^{1}H δ; 9.2(s), 7.8(s), 7.7(s), 7.4(s), 5.4(s) ^{13}C δ; 129.0, 120.8, 52.0	NMR: ^{1}H δ; 9.4(s), 7.8(s), 7.5(s), 7.4(m) 5.5(s) ^{13}C δ; 135.5, 129.0, 122.4, 120.6, 51.9	NMR: ^{1}H δ; 9.3(s), 7.8(s), 7.7(s), 7.4(s), 5.4(s) ^{13}C δ; 135.8, 129.4, 120.7, 52.1

Table 7c: IR, Raman, and NMR Data

	NO_3^-	SO_4^{-2}	BF_4^-	Br^-	$CF_3SO_3^-$
2-Methyl-imidazolium	IR: 3111, 1576, 1388, 1078, 711 cm^{-1}	IR: 3190, 3005, 2650, 1185, 850, 610 cm^{-1}	IR: 3450, 3060, 1700, 1050, 865 cm^{-1}	IR: 3420, 2950, 2670, 1300, 840 cm^{-1}	IR: not available
	Raman: 1510, 1110, 1042, 670, 150 cm^{-1}	Raman: 1105, 983, 611, 450 cm^{-1}	Raman: 762, 730, 668 cm^{-1}	Raman: not available	Raman: not available
	NMR: ^{1}H δ; 7.5(s)	NMR: ^{1}H δ; 9.3(s), 7.3(s), 2.5(s)	NMR: ^{1}H δ; 7.5(s), 2.5(m)	NMR: ^{1}H δ; 7.5(s), 3.4(s)	NMR: ^{1}H δ; 7.5(s), 3.9(s), 2.6(s)
	^{13}C δ; 144.5, 118.9, 11.4	^{13}C δ; 119.9, 12.4	^{13}C δ; 144.5, 118.9	^{13}C δ; 144.0, 118.9,	^{13}C δ; 144.5, 118.9, 11.4

Table 7d: IR, Raman, and NMR Data

	NO_3^-	SO_4^{-2}	BF_4^-	Br^-	$CF_3SO_3^-$
2-Ethyl-imidazolium	IR: 3400, 3150, 1640, 1410, 1050, 930cm^{-1}	IR: 2810, 2000, 1680, 1040, 985, 510 cm^{-1}	IR:not available	IR: 3010, 2740, 810 cm^{-1}	IR: not available
	Raman: 1511, 1116, 1043, 643, 136 cm^{-1}	Raman: 1105, 983, 611, 450 cm^{-1}	Raman: 1070, 777, 533, 360 cm^{-1}	Raman: 1510, 1123, 957, 917 cm^{-1}	Raman: 1516, 1127, 1034 cm^{-1}
	NMR: ^{1}H δ; 7.6(s), 2.9(m), 1.3(m), 0.9(s)	NMR: ^{1}H δ; 9.4(s), 7.2(s), 2.8(m), 1.3(m)	NMR: ^{1}H δ; 7.5(s), 2.9(m), 1.3(m)	NMR: ^{1}H δ; 3.5(s), 2.9(m), 2.8(s), 1.3(s),	NMR: ^{1}H δ; 10.6(s), 7.5(s), 2.9(m), 1.3(m)
	^{13}C δ; 148.9, 118.9, 19.2, 11.6	^{13}C δ; 149.1, 119.9, 20.1, 12.3	^{13}C δ; 149.1, 118.8, 19.1, 11.4	^{13}C δ; 116.4, 37.6, 9.14	^{13}C δ; 149.1, 119.2, 19.4, 11.6

Conclusion

Materials research and development for nonlinear optical applications has progressed to the point where several systems are commercially available. So far, no material system has proven to possess all the desired properties for nonlinear optics. Thus, considerable research is still needed to develop materials that can fulfill the critical requirements of devices used in information processing, optical frequency conversion, integrated optics, and telecommunications.

This project was limited to a specific salt series based around the 1, 3-disubstituted imidazolium cation. However, given the fact that there are virtually limitless numbers of permutations of cations and anions this project could lead to a massive growth in the number of known SHG active materials. Based on the positive results achieved in this preliminary study, work is continuing on extending the salt series to improve the NLO activity and the desired structural properties. In addition, improvements on the currently available SHG apparatus are planned which would allow quantitative measurements to be recorded. A separate project using X-ray crystallography to evaluate the structure of these salts is also underway.

References

1. Boyd, R.W.; <u>Nonlinear Optics;</u> Chapter 1, Academic Press, San Diego, Ca 1992.

2. Perry, J.W.; In <u>New Materials for Nonlinear Optics</u>; Marder, S.R., Sohn, J.A., Stucky, G.D., Eds.; ACS Symposium Series No. 455, American Chemical Society: Washington, D.C., 1991; p. 67.

3. Bloemberger, N.; <u>Nonlinear Optics</u>; Benjamin Press, New York, NY 1965.

4. Greene, B.I., Orenstein, J., Schmitt-Rink, S.; <u>Science</u>; vol. 247, 1990, p. 679.

5. Marcy, H.O., Warren, L.F., Webb, M.S., Ebbers, C.A., Velsko, S.P., Kennedy, G.C., Catella, G.C.; <u>Applied Optics</u>; vol. 131, no. 24, 1992, p. 5051.

6. Eaton, D.F. In <u>New Materials for Nonlinear Optics</u>; Marder, S.R., Sohn, J.A., Stucky, G.D., Eds.; ACS Symposium Series No. 455, American Chemical Society: Washington, D.C., 1991; p. 128.

7. Eaton, D.F.; <u>Chemtech</u>; May 1992, p. 308.

8. Stoddart, J.F.; <u>Journal of the American Chemical Society</u>; vol. 114, 1992, p. 193; Sakaguchi, H., Nagamura, T., Matsuo, T.; <u>J. Chem. Soc. Chem. Commun.</u>; 1992, p. 209.

9. Eaton, D.F.; <u>Science</u>; vol. 253, 1991, p. 281.

10. Cheng, L.T., Tam, W., Stevenson, S.H., Meredith, G.R., Rikken, G., Marder, S.R.; <u>J. Physical Chem.</u>; vol. 95, no.26, 1991, p. 10643.

11. Marder, S.R., Perry, J.W., Schaeffer, W.P.; <u>Science</u>; vol.245, 1989, p. 626.

12. Warren, L.F. In <u>Electronic Materials-Our Future</u>; Allred, R.E., Martinez, R.J., Wischmann, K.B., Eds.; 4th International SAMPE Electronics Conference vol. 4, June 12-14, 1990, p. 388.

13. Kurtz, S.K., Perry, T.T., <u>J. Applied Physics</u>; vol. 39, 1968, p. 3798.

AN ARTIFICIAL NETWORK APPROACH TO STRUCTURAL DAMAGE DETECTION USING FREQUENCY RESPONSE FUNCTIONS

Clinton R. Povich
Graduate Student
Tae W. Lim
Assistant Professor
Department of Aerospace Engineering

University of Kansas
2004 Learned Hall
Lawrence, KS 66045

AN ARTIFICIAL NEURAL NETWORK APPROACH TO STRUCTURAL DAMAGE DETECTION USING FREQUENCY RESPONSE FUNCTIONS

Clinton R. Povich
Graduate Student
Tae W. Lim
Assistant Professor
Department of Aerospace Engineering
University of Kansas

Abstract

Structural damage detection in a twenty-bay planar truss was accomplished using an artificial neural network. Instead of using natural frequencies and mode shapes, the frequency response functions (FRF's) experimentally obtained from accelerometers at two locations on the truss were directly used to distinguish among damage cases and to train the network. Unlike conventional approaches based on system identification techniques, the neural network approach does not require an analytical model of the structure. The direct use of the FRF's eliminates the need for modal parameter identification. Out of the 60 damage cases considered in this study, the neural network was able to identify uniquely 21 damage cases and narrowed 38 others down to two possible damaged struts.

AN ARTIFICIAL NEURAL NETWORK APPROACH TO STRUCTURAL DAMAGE DETECTION USING FREQUENCY RESPONSE FUNCTIONS

Clinton R. Povich
Tae W. Lim

Introduction

To maintain the performance and safe operation of aerospace vehicles, structural integrity must be monitored periodically. Numerous studies have been conducted to detect structural damage using measured mode data and system identification techniques [1-4]. Recently, attention was given to the application of the artificial neural network to structural damage detection [5,6]. The natural frequencies and mode shapes are used to train the network and to conduct damage detection. Unlike the conventional approaches based on system identification techniques, the neural network approach does not require a refined model of the undamaged structure. The fact that damage detection can be performed using the measured mode data alone without an analytical model relieves the burden of conducting analytical model refinement to match measured mode data, which is typically costly and time consuming.

This paper investigates an effective use of artificial neural network to conduct structural damage detection. In a typical modal test, the modal parameters are identified using the frequency response functions (FRF's). Instead of using mode shapes and frequencies, the neural network can be trained directly using the FRF's. Some of the advantages using the FRF's directly include: (1) the modal parameter identification process is not required and thus the error and cost associated with it are eliminated, (2) any nonlinear behavior exhibited in FRF's can be incorporated in the damage detection process, and (3) the number of transducers required to perform damage detection can be significantly reduced as will be shown in the twenty-bay truss structure example.

Test Apparatus and Procedure

The focus of this study is damage detection in a twenty-bay two-dimensional truss (see Fig. 1). The 7.07 m long truss fixed at one end and free at the other lies flat on a table. The truss consists of Meroform hollow aluminum tubes and steel node balls. Underneath each batten, a steel bar is bolted to the node balls. Each steel bar rests on two steel ball bearings that allow the truss to move in the horizontal plane. Detailed information on the truss is available in Ref. [7]. At the end of the truss, a MB Dynamics Modal 50 Exciter (shaker) is attached to provide force excitation. Accelerometers are located at the numbered positions as shown in Fig. 1. Since the minimum number of accelerometers needed was unknown, eleven accelerometers were installed to provide sufficient modal test data. Then, the minimum number of accelerometers and their locations were to be determined by inspecting the FRF's and evaluating the damage detection performance of the trained neural network. Consequently, only the FRF's from numbers 5 and 11 are selected to train the neural network and to conduct damaged detection. Accelerometer 11 is chosen because the FRF's at this point displayed the greatest diversity. The other accelerometer was used to provide uniqueness to the few damage cases that had similar FRF's from accelerometer 11.

A different strut is removed for each of the 60 damage cases investigated to simulate complete damage in that strut. In each bay, only the diagonals and longerons are removed. It is difficult to remove the battens because of the steel bars connecting the nodes of each batten. Thus, they are eliminated from damage consideration in this study. The FRF's are then found using the Tektronix 2630 Fourier Analyzer. The Fourier Analyzer controls the random force input to the shaker and records the FRF's over a range of 0 to 50 Hz, which were obtained using 20 averagings. The FRF's obtained for all damaged and undamaged cases contain at least four bending modes below 50 Hz. Figures 2 and 3 show the FRF's measured from accelerometers 5 and 11, respectively. The figures show the changes in FRF's due to damage in upper longerons, diagonals, and lower longerons. Each figure has 21 FRF's corresponding to the undamaged case and the damage cases in bay 1 through 20. Thus, the FRF corresponding to damage case number

14-4

10 for the upper longeron case in Fig. 2 indicates the FRF measured at accelerometer 5 with upper longeron removed in bay 9. Figure 4 shows in detail the FRF of a typical damage case plotted with the FRF of the undamaged case at accelerometer 11.

The sample rate employed to measure the FRF's was 2.56 times the bandwidth (50 Hz) and 512 samples were taken for each test. Since only data up to 50 Hz is desired, each FRF consists of 200 frames (201 data points). Of these 201 points, the first four are not included in the neural network training due to the large 'spike' in the data at these points in nearly every FRF of every damage case (see Fig. 4). The resulting FRF's then have a range of 1.25 to 50 Hz (197 data points). Finally, the FRF data is converted to a Matlab file to conduct neural network training.

Artificial Neural Network

The Matlab Neural Network Toolbox [8] was used to construct and train the neural network. The type of network used in this study was a tree-layer neural network using log-sigmoid neurons. The backpropagation method was used to train the network.

A neural network is composed of one or more layers and each contains a series of processing units (neurons) that act in parallel within the layer. Data arranged in a vector (or a matrix of vectors) is transferred through the layers. Each data value within a vector is multiplied by the weight associated with each neuron in a layer and the results is added to the neuron's bias. Then, the results are summed at each neuron creating a new vector with a size equal to the number of neurons in that layer. A transfer function converts this new data set and sends it to the next layer. The transfer function used in this study is a log-sigmoid which converts any real number to a value between 0 and 1. In the target output for the network, a '1' signified no damage in a strut while a '0' indicated complete damage (missing).

Initially, the weights and biases for all of the neurons are randomly chosen. After one pass (epoch) through the network, the output matrix is compared to the target output matrix. The neural network computes the delta error vectors (derivatives of error) which are backpropagated through

the network until delta vectors are found for each layer. The learning rate is then used to change

the weights and biases in the direction of steepest decent with respect to the delta vectors. For

example, the change in a weight, delW, and a bias, delB, are found as

$$delW(i,j) = lr*D(i)*P(j) \tag{1}$$

$$delB(i) = lr*D(i) \tag{2}$$

where

lr = learning rate

D(i) = delta vector of a layer

P(j) = input vector to a layer

i = number of neurons in layer

j = number of elements in input vector to a layer

The inputs are sent through the network layers again and the process is repeated. This process is

referred to as training. Training stops when the sum of the squared errors for each training vector

is less than the error goal specified by the user. In this study, the error goal is selected as 1e-10 to

confine individual errors within about ±1e-6.

An adaptive learning rate allows the network to be trained faster by comparing the new

error vector after each epoch with the previous (old) error vector. If the new error vector is larger

than the old (by 1.04), the new weights and biases are discarded and the learning rate decreases by

0.7. If the new error vector is less than the old, the learning rate is increased by 1.05 and the new

weights and biases are kept. Also, momentum is used such that the network avoids getting stuck

in a local minimum of the overall error surface. Basically, momentum allows the network to

ignore small changes and respond to the general trend in the error surface. A momentum constant

is chosen before training with a value between 0 and 1. The quantity 0 signifies a weight change based solely on the gradient while 1 sets the new weight change equal to the last weight change and the gradient is ignored, i.e.,

$$delW(i,j) = mc*delW(i,j) + (1 - mc)*lr*D(i)*P(j) \tag{3}$$

where the quantity mc is a momentum constant. A typical value is 0.95.

For the network in this study, 61 input vectors are used to train the network: 60 damage cases and 1 undamaged case. Each input vector contains the FRF's with the FRF from accelerometer 5 listed first than that of accelerometer 11. Thus, the size of the input matrix becomes 394 by 61. Two hidden layers of 125 and 40 neurons, respectively, are included in the network to achieve the desired error goal with a reasonable amount of training time. The target output vector corresponding to each input vector was 60 element vector consisting of 1's (denoting struts with no damage) and a 0 (damaged strut). The undamaged case contains all 1's. Therefore, the size of the target matrix is 60 by 61. Each element in the target vector corresponds to a truss strut with following pattern

Element		Strut description
1		upper longeron
2	Bay 1	diagonal
3		lower longeron
4		upper longeron
5	Bay 2	diagonal
6		lower longeron
...		...
58		upper longeron
59	Bay 20	diagonal
60		lower longeron

The planar truss structure that is used to perform damage detection exhibits interesting dynamic characteristics mainly due to the lacing pattern of the diagonals. Each longeron does not produce unique changes in the FRF's. The two longerons located at the open end of each 'V' created by two diagonals in adjacent bays produced nearly identical FRF's when either was removed (see Fig. 5). This resulted in 19 pairs of longeron damage cases with indistinguishable dynamic characteristics. Therefore, the best that can be achieved for longeron damage cases was too narrow down from 60 damage candidates to two adjacent longerons. Also, when the lower longeron in the 20th bay is removed the resulting FRF is similar to that for the undamaged case; thus making it difficult to detect damage in that strut (see Fig. 6). To accommodate these characteristics of the truss structure, the target output vectors are modified as follows: for the 19 longeron pairs, each vector contains two 0's representing the two possible damage cases and for the 20th lower longeron, no 0's are present. All the diagonals produced dynamic characteristics that are uniquely identifiable. The number of epochs taken for the network described above to be trained was between 1200 to 1500 when random initial weights and biases were used. Figure 7 shows the error distribution between the target output and actual output computed using the final weights and biases of the trained network. Most of individual errors are confined within ±1e-6 thanks to the small summed squared goal of 1e-10.

To evaluate the performance of the trained network, the FRF's of several damage cases which were not used to train the network, were presented to the network. Figure 8 shows the plot of the resulting output vectors of several damage cases. To clarify the location and magnitude of damaged elements, each element of the output vector was subtracted by 1. Thus, the quantities 0 and 1 represent no damage and complete damage, respectively, in the figure. The location and the magnitude of damage is clearly indicated in the plot. In nearly all damage cases, the location and magnitude of damage was clearly identified. For longeron damage cases, two struts are identified because of the reasons discussed previously.

Conclusions

Successful damage detection was accomplished using the artificial neural network trained with experimentally obtained FRF's from just two accelerometers on a two-dimensional truss. The neural network successfully located a missing strut in 21 cases out of the 60 damage cases investigated. The 21 cases include all diagonal damage cases and the lower longeron damage in the first bay. In 38 of the 60 damage cases, the network narrowed the 60 candidates down to two adjacent longerons. For one case, which corresponds to the lower longeron in the 20th bay, it was not successful in locating the missing strut. The success of damage detection was directly tied to the uniqueness of the FRF that was produced by a certain damage case. For instance, due to the lacing pattern of the diagonals of the truss, 19 longeron pairs produced almost identical FRF's. Thus, pinpointing the damaged member was impossible.

A neural network trained using natural frequencies and/or mode shapes may have produced the same results but the number of accelerometers required would have been at least four to be able to distinguish four bending modes below 50 Hz. The direct use of FRF's investigated in this report eliminates the need for the modal parameter identification process and helps avoid the cost and error associated with it. Also, a refined analytical model is not required unlike the conventional approaches based on system identification techniques. Thus, the burden of performing modal refinement to obtain high fidelity analytical model is removed.

Further study should include the response of the trained network to partially damaged struts and/or combinations of damaged struts and the need to include these in training. Additional FRF's from other accelerometers may also be needed for these cases. The FRF of an axially mounted accelerometer may help separate the indistinguishable pairs of damage cases. A similar study on a three-dimensional truss might prove more successful since the number of damage cases that produce almost identical FRF's may be fewer.

References

1. Smith, S. W. and Hendricks, S. L., "Damage Detection and Location in Large Space Trusses," AIAA SDM Issues of the International Space Station, A collection of Technical Papers, Williamsburg, VA, 1988, pp. 56-63.

2. Kashangaki, T., "On-Orbit Damage Detection and Health Monitoring of Large Space Trusses - Status and Critical Issues," AIAA Paper 91-1181, April 1991.

3. Lim, T. W., "Structural Damage Detection Using Modal Test Data," AIAA Journal, Vol. 29, No. 12, 1991, pp. 2271-227.

4. Zimmerman, D. C. and Kaouk, M., "Structural Damage Detection Using a Subspace Rotation Algorithm," AIAA Paper 92-2521, April 1992.

5. Tsou, P. and Shen, M.-H. H., "Structural Damage Detection and Identification Using Neural Network," AIAA Paper 93-1708-CP, April 1993.

6. Yen, G. G. and Kwak, M. K., "Neural Network Approach of the Damage Detection of Structures," AIAA Paper 93-1485-CP, April 1993.

7. Hallauer, W. L. and Lamberson, S. E., "A Laboratory Planar Truss for Structural Dynamics Testing," Experimental Techniques, Vol. 13, No. 9, 1989, pp. 24-27.

8. Demuth, H. and Beale, M., Neural Network Toolbox User's Guide, MathWorks, Inc., Natick, Massachusetts, 1992.
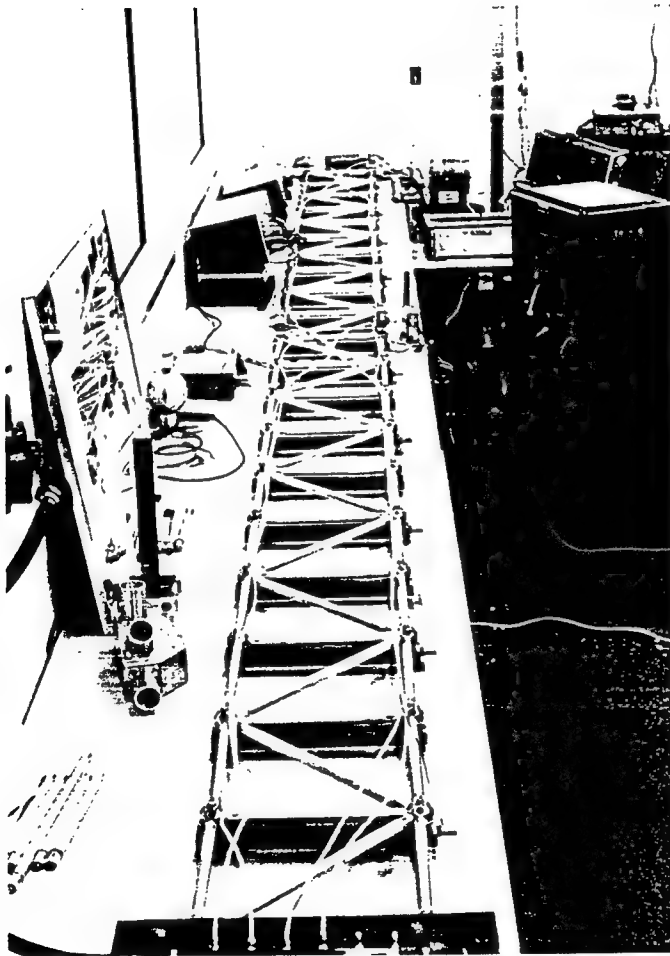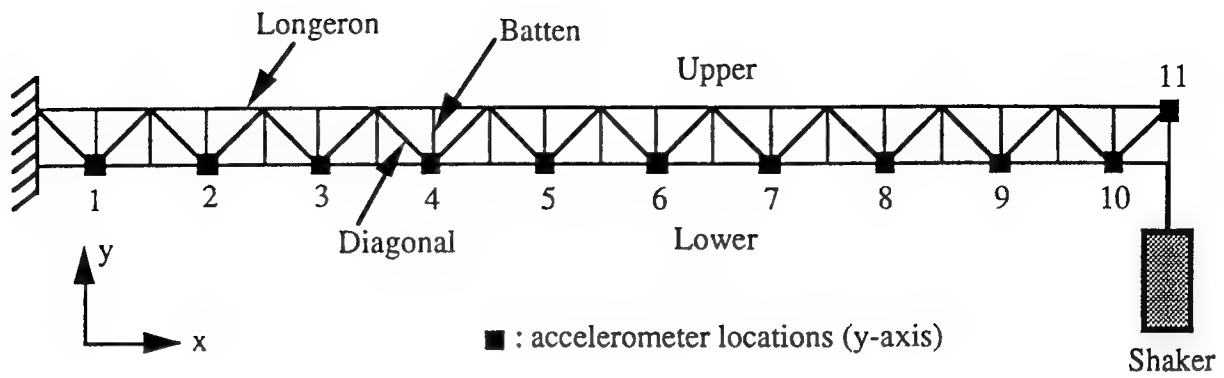
Fig. 1 A twenty-bay planar truss structure used for damage detection
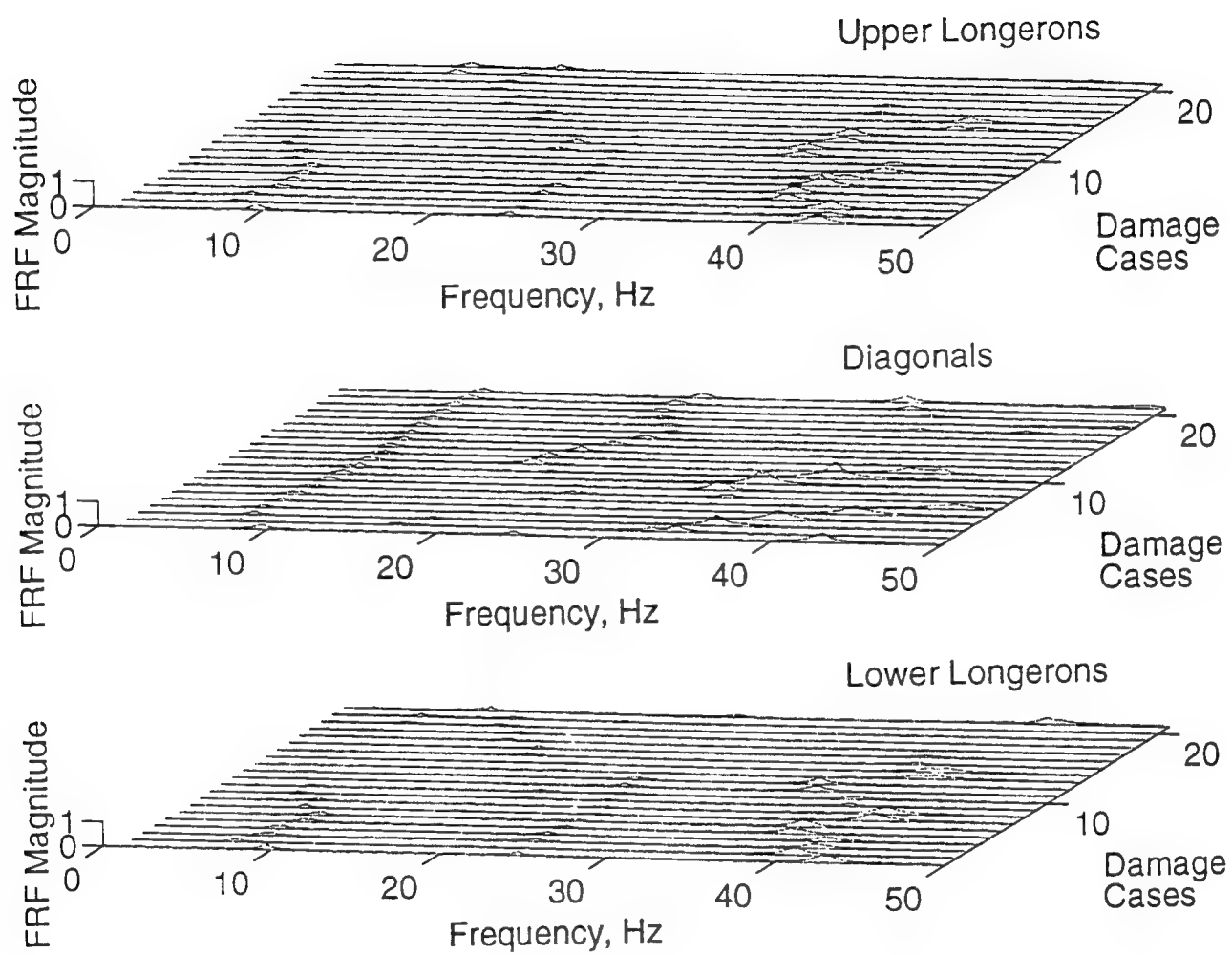
Fig. 2 FRF's measured at accelerometer 5
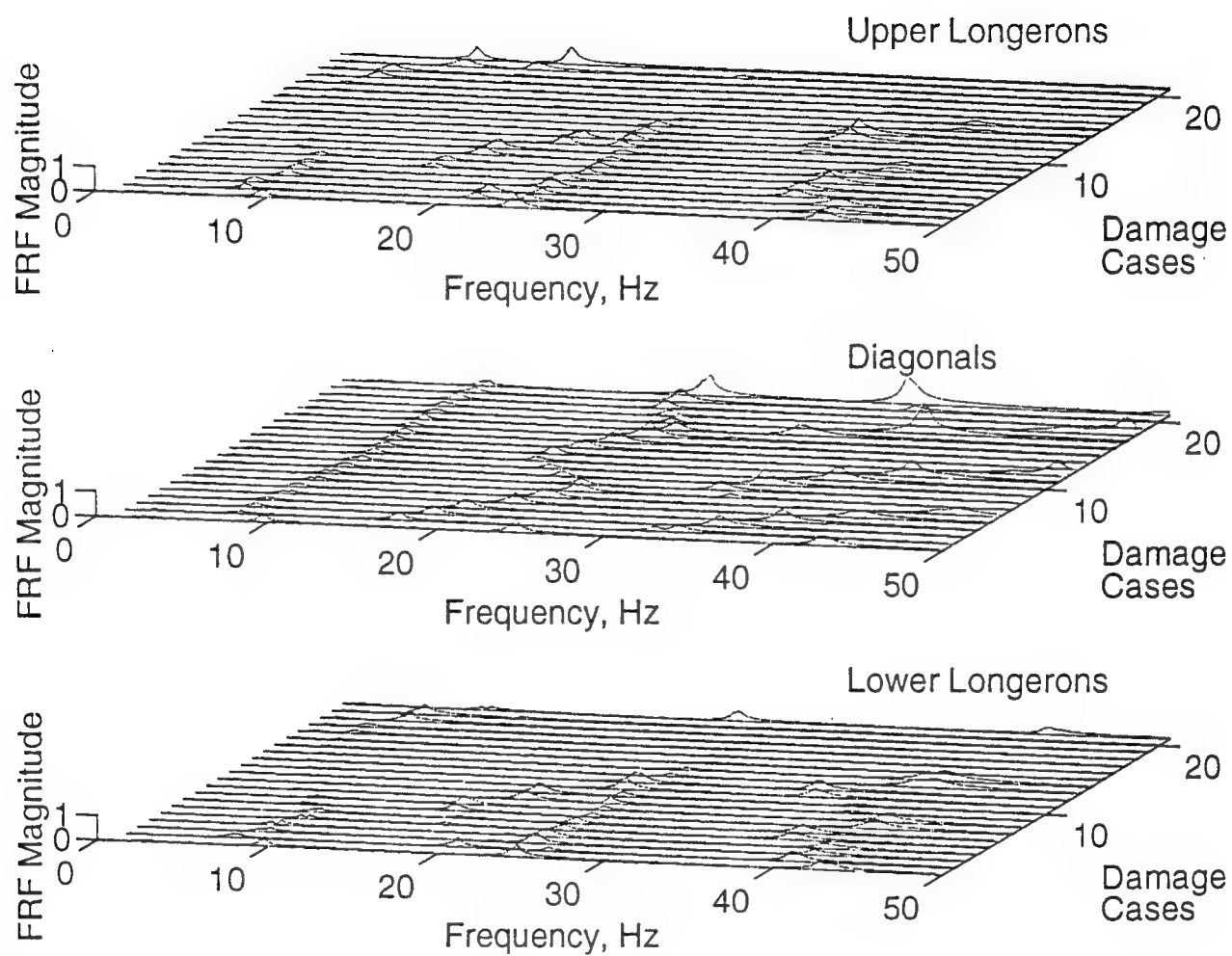
Fig. 3 FRF's measured at accelerometer 11
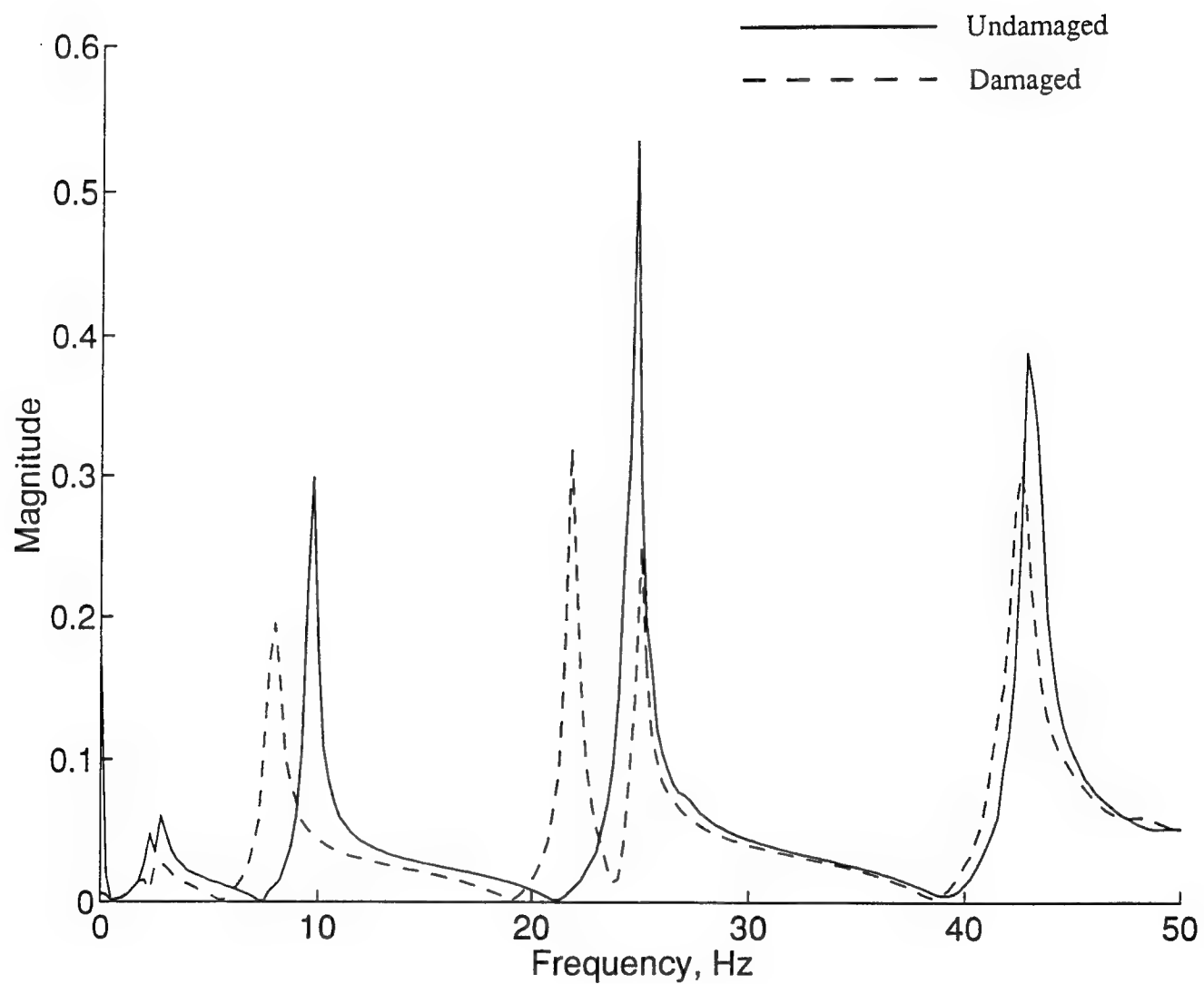
Fig. 4 Comparison of the undamaged and a typical damaged truss FRF's at accelerometer 11
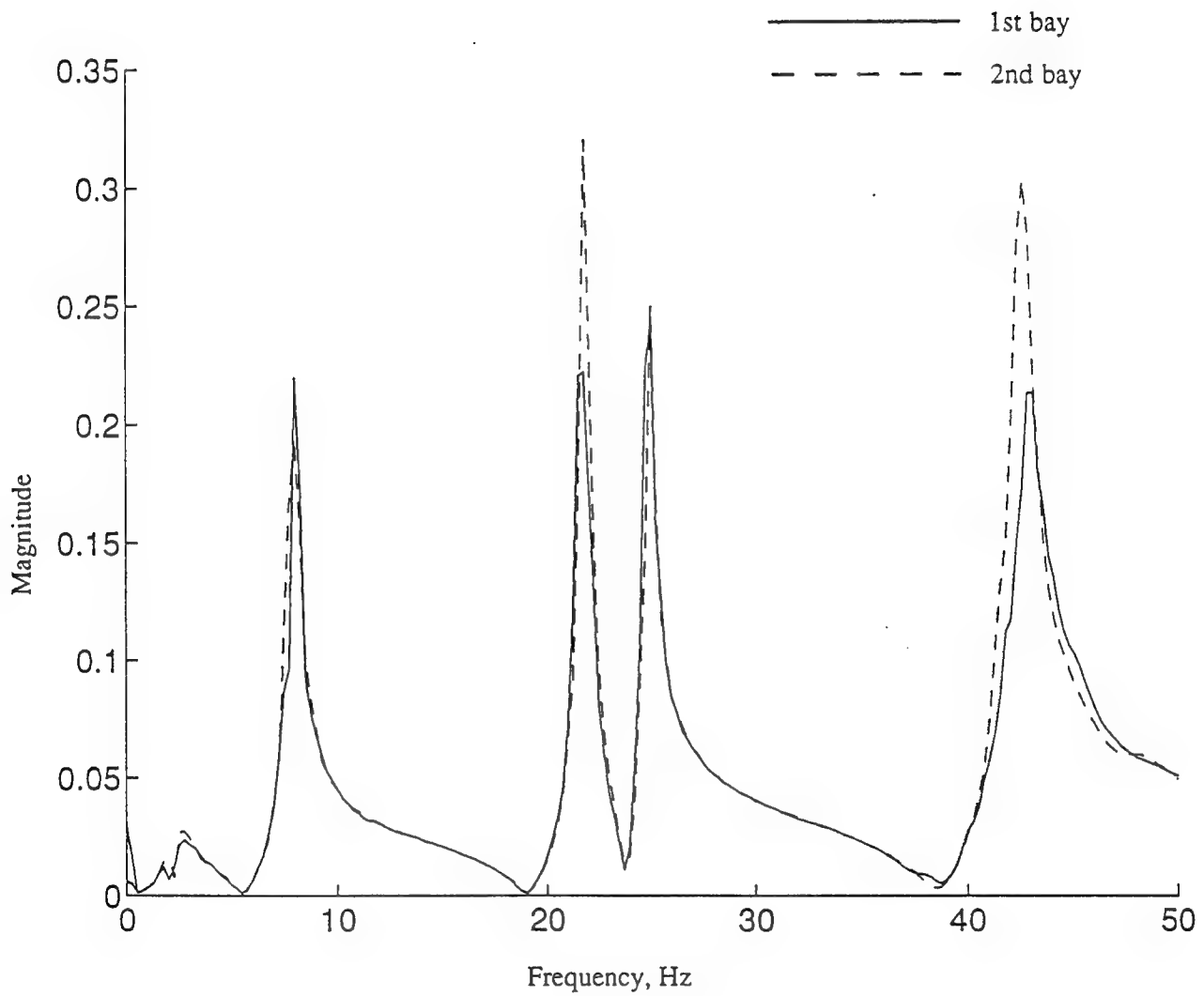
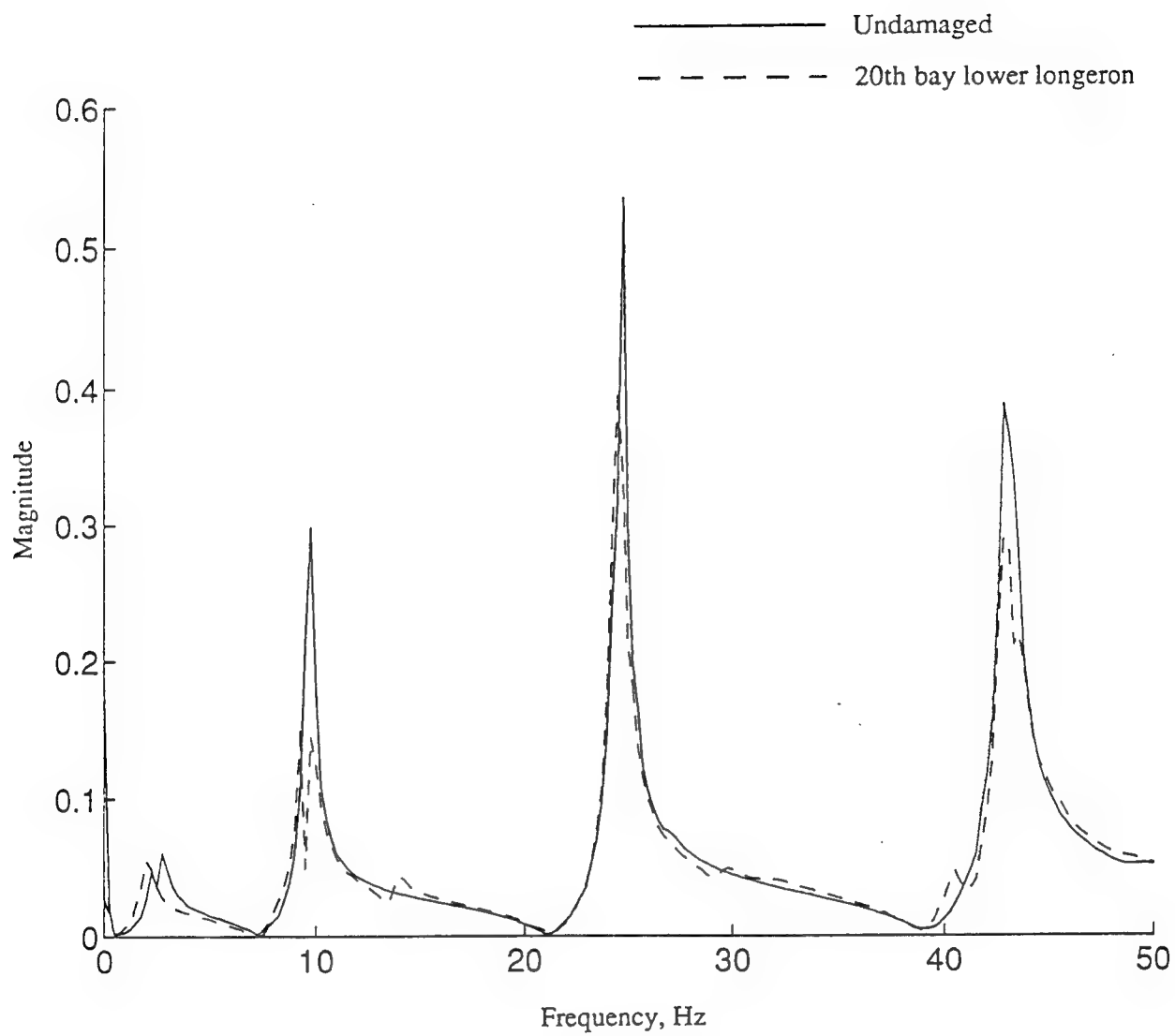Fig. 5 FRF's at accelerometer 11 for missing upper longerons in the 1st and 2nd bays

Fig. 6 FRF's at accelerometer 11 for the undamaged truss and missing lower longeron
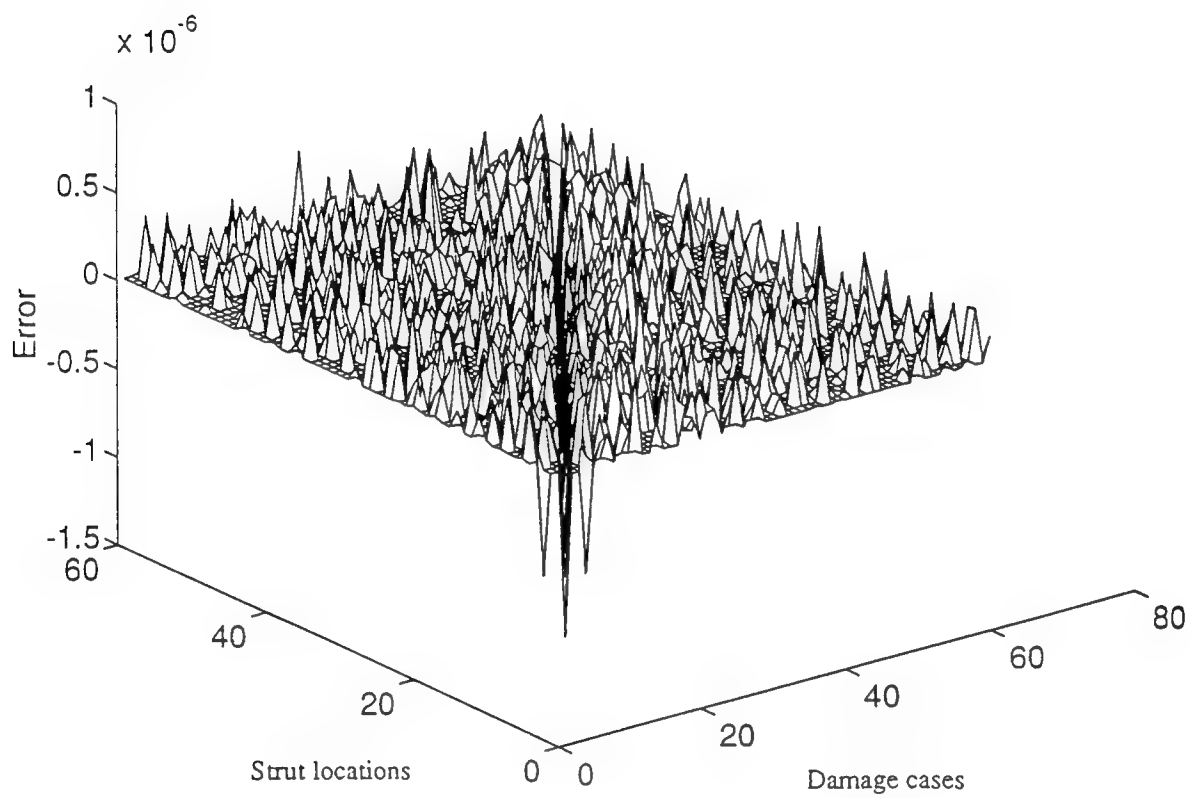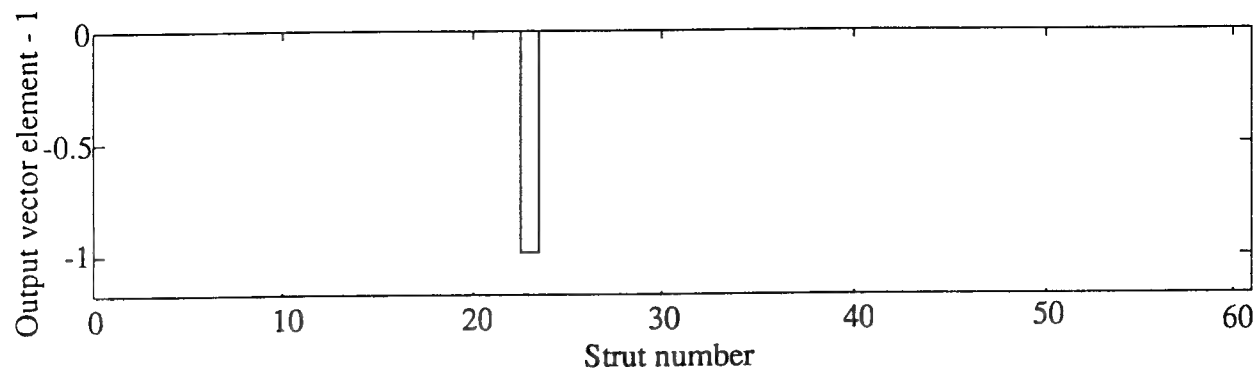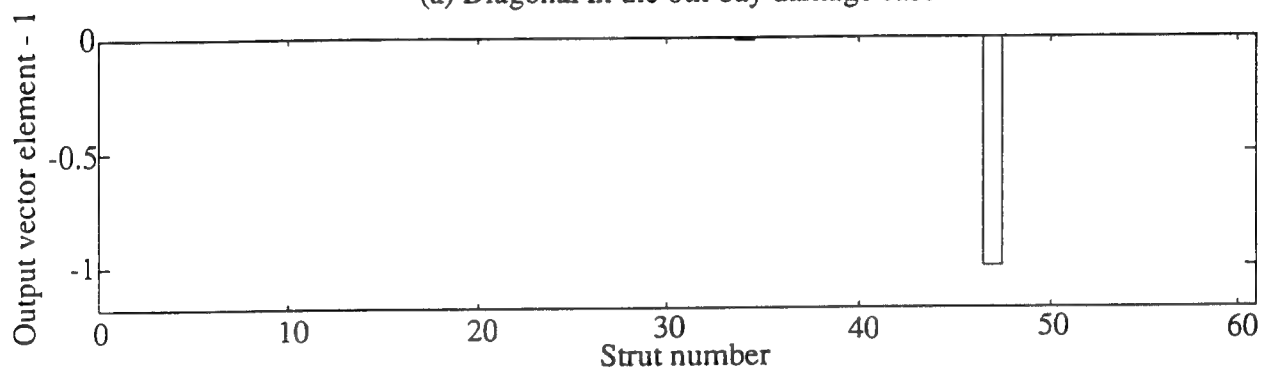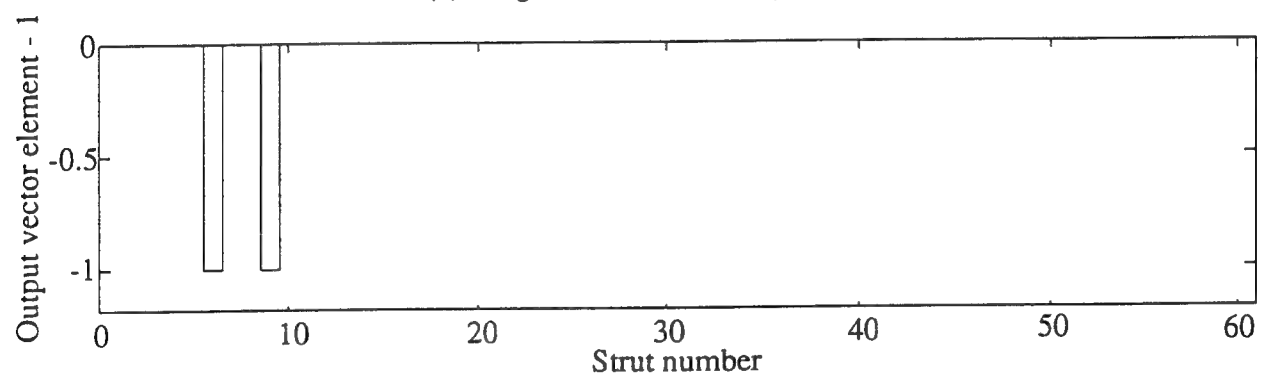in the 20th bay

Fig. 7 Error distribution of the trained network

(a) Diagonal in the 8th bay damage case

(b) Diagonal in the 16th bay damage case

(c) Lower longeron in the second bay damage case
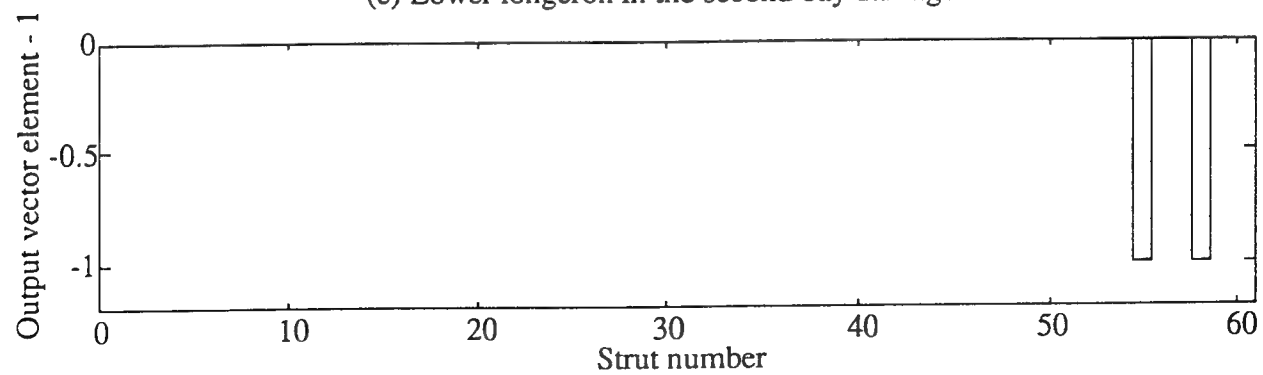
(d) Upper longeron in the 19th bay damage case

Fig. 8 Damage detection results using the trained neural network

ELECTRIC FIELD INDUCED SECOND HARMONIC GENERATION IN GERMANIUM DOPED

SILICA PLANAR WAVEGUIDES

Peter S. Weitzman

Ph.D. Candidate

Thayer School of Engineering

Dartmouth College

Hanover, NH 03755

15-1

ELECTRIC FIELD INDUCED SECOND HARMONIC GENERATION IN GERMANIUM DOPED SILICA PLANAR WAVEGUIDES

Peter S. Weitzman
Ph. D. candidate
Thayer School of Engineering
Dartmouth College

Abstract

Frequency doubling in germanium doped silica planar waveguides deposited on fused silica substrates is studied. It is demonstrated that an externally applied, periodic DC field can cause instantaneous frequency doubling in these waveguides. The periodicity which causes frequency doubling corresponds to the beat length between fundamental and second harmonic light propagating in the waveguide. A current has been measured from these periodic electrodes which corresponds to the generation of a periodic DC field internal to the glass when illuminated with both fundamental and second harmonic light.

ELECTRIC FIELD INDUCED SECOND HARMONIC GENERATION IN GERMANIUM DOPED

SILICA PLANAR WAVEGUIDES

Peter S. Weitzman

Introduction

Efficient frequency doubling of 1.064nm radiation was first observed in germanium doped silica fibers in 1986 [1]. Recently this has also been achieved in planar dielectric waveguides made from the same materials [2]. Second harmonic generation (SHG) in amorphous materials is interesting due to the fact that the second order susceptibilty, $\chi^{(2)}$, is zero. There have been several theories proposed to explain this phenomenon [3,4,5,6]. All of these theories are based on Stolen and Tom's original hypothesis that an internal DC electric field is responsible for an effective $\chi^{(2)}$ and that the periodicity of this field provides the quasi phase matching necessary for efficient SHG [7]. Assuming that phase matching has been achieved, the conversion efficiency is dependent on the overlap integral between the fundamental and second harmonic optical fields and the DC electric field, which is given by the equation

$$\eta \propto \left| \chi^{(3)} \iint E_{DC} E_{2\omega} E_{\omega} E_{\omega} dA \right|^2 \qquad (1)$$

where η is the conversion efficiency and the integral is over the transverse cross section of the waveguide. This effect has been studied extensively in fibers, however the planar waveguide geometry provides easy access to the core region of the waveguide where this internal electric field can be modified and probed.

In 1975, Levine and Bethea [8] applied a periodic electric field to a liquid planar waveguide. In 1989, Kashap [9] applied an external periodic DC field to the exposed core of an optical fiber and

was able to induce SHG. In this paper we present results of using a similar technique on a planar dielectric waveguide. We have applied a periodic electric field to the planar waveguide and observed SHG and have also applied both 1.064 micron (IR) and 0.532 micron (green) radiation simultaneously to a planar waveguide and observed the buildup of the internal electric field by monitoring the current in the periodic electrode structure.

In order to provide the correct periodicity for quasi phase matching, the nonlinearity must be strong only at points where the fundamental and second harmonic are in phase. This periodicity, Δz is given by the expression

$$\Delta z = \left| \frac{2\pi}{2\beta_\omega - \beta_{2\omega}} \right| \tag{2}$$

where β_ω is the propagation constant for the IR light and $\beta_{2\omega}$ is the propagation constant for the green light. The planar waveguide geometry provides a simple means for determining these propagation constants by prism coupling. Figure 1 Illustrates the geometry of the planar waveguide and coupling prisms. The propagation constants are determined from the internal prism angles by the equation

$$\beta = n_p k_0 \cos\theta_p \tag{3}$$

where n_p is the prism refractive index, θ_p is the incidence angle of the light internal to the prism and k_0 is the optical wavenumber in free space. These propagation constants can also be determined from knowledge of the film thickness and refractive index by solving the appropriate eigenvalue equation [10]. The same eigenvalue equation can also be used to determine the film

15-4

thickness and index of refraction from the observed coupling angles for a number of modes in the waveguide.

Methodology

The waveguides used to obtain the results in this paper were fabricated using argon ion beam sputtering of silica and germanium onto commercial, optical grade fused silica substrates. The sputtering rate was 0.11nm/sec in a partial pressure of 10^{-4} Torr of oxygen. The two films used in this study were approximately 2.0 and 3.5 microns thick. Auger analysis indicated that the GeO_2 content of the films was approximately 6m%. The refractive index of the substrate is 1.4607 at 0.532μm and 1.4496 at 1.064μm. The refractive index of the film was measured to be 1.546 at 0.532 μm and 1.532 at 1.064μm using the prism coupling method.

Figure 2 illustrates the geometry of the planar waveguide used along with the periodic electrode structure mounted on the surface of the waveguide. This waveguide/electrode combination is mounted in the setup illustrated in figure 3. Some of the output from a Q switched, mode locked Nd:YAG laser is frequency doubled in a KTP crystal, subsequently the Fundamental and SH beams are separated into two paths. Using this technique, the input coupling angle of the two wavelengths can be adjusted independently, as can their power levels, polarization and relative phase. The disadvantage of separating the beams is that the slight relative phase fluctuations between the paths can wash out the grating as it is forming using the seeding technique [11]. Despite these phase fluctuations, we have still been able to prepare these films for SHG. The output SH signal is detected with a photomultplier and averaged with a boxcar integrator. Typically 3000 q-switched pulses are averaged and read by the computer.

We have performed two different types of experiments. In the first experiment, the voltage source is applied to the electrode structure, only IR light is coupled through the waveguide and the angle of

the electrodes with respect to the optical path is varied in order to optimize the detected SH signal. By adjusting the angle of the electrodes, the effective periodicity can be modified. Once the SH signal is optimized, the SHG is measured as a function of applied voltage.

In the second experiment, the optimized electrodes are kept in place but they are connected to an electrometer capable of measuring current as low as 100aA. The waveguide/electrode assembly and associated wires are placed in a metal box to shield them from electromagnetic interference. At this point, both IR and green light can be coupled simultaneously into the waveguide. Care was taken to insure that they travel along the same region of the waveguide. The current induced in the electrodes is measured by the electrometer as a function of the propagation of various optical frequencies within the waveguide.

Results

Using equation (2) it was determined that the quasi phase matching period for the 3.5μm waveguide is 26.7 μm and for the 2μm waveguide it is 20.2 μm. This was for the lowest order TM mode in both the green and IR. The TM modes were used due to the fact that more efficient SHG has been observed in these modes. (2) We chose to use the lowest order modes because they have the largest overlap integrals with the applied DC field. We found the greatest SHG on the 3.5 micron thick waveguide using an electrode structure with a 26μm periodicity rotated 7.9 degrees to the optical path. This provides an effective electrode periodicity of approximately 26.25μm, very close to what we calculated using equation (1). The maximum SHG with 50 volts applied across the electrodes was a visible green spot at the correct angle to indicate that it was in the TM_0 mode. The power produced by the SHG was strongly dependent on the periodicity of the waveguide. A rotation of less than 1 degree from the optimum caused a 90% decrease in the SHG produced. With the electrode periodicity fixed at 7.9 degrees the SHG was measured as a function of applied voltage

(Fig 4). The observed quadratic dependence of the electric field induced SHG is expected due to the dependence of the conversion efficiency on the square of the electric field in equation 1 .

We were also able to generate a signal in the TM_2 mode in this film by changing the electrode periodicity to 70μm, close to the calculated Δz value for this mode. The data for SHG in the TM_2 mode vs applied voltage is plotted in figure 5. It is several orders of magnitude lower than what was observed for the TM_0 mode. We also tried to generate the TM_1 mode but were unable to see any signal. This is expected due to the cancellation in the overlap integral. We also tried to produce SHG with the input light in the TE_0 mode, and the electrodes aligned to produce quasi phase matching between the TE_0 modes in IR and green. For this configuration, no SHG was observed. This is most likely due to the fact that the applied electric field was almost perpendicular to the electric field of the optical modes.

With the electrodes optimized for quasi phase matching the TM_0 mode, we disconnected the voltage source and connected the electrometer to the electrodes. We then measured the current induced into the electrodes with both the IR and green light coupled into the film, the IR only, the green only, and no light. The measured current changed with time at a rate which was dependent on the coupling of light through the waveguide. With no light present, an increasing current was detected, due to the discharging of the electrodes through the electrometer. When a combination of both IR and green light was coupled through the waveguide, the current was obsevred to decrease. When only green or IR alone was coupled in, the current increased. This data is plotted in figure 6.

Discussion

We have presented a device consisting of a germanium doped glass waveguide and a periodic electrode structure which is capable of instantaneously frequency doubling light. We have demonstrated that a periodic DC electric field is capable of inducing this frequency doubling in glass

15-7

if the periodicity is correct for quasi phase matching. We have been able to selectively excite different modes in the SH by varying the electrode periodicity. The efficiency of the electric field induced SHG is greatly reduced for higher order modes. This can be explained by examining the overlap integrals involved. The conversion efficiency is determined by the overlap integral in equation 1.

The three lowest order modes and the electric field distribution in the waveguide are illustrated in figure 7. From this figure, it is observed that the positive and negative lobes in the higher order modes will cancel out when integrated across the applied electric field. The applied field does decrease as the distance from the electrodes is increased. This will result in a slightly nonzero overlap integral for the first order mode, however the separation between the electrodes is much larger than the film thickness so the change in electric field across the film is less than 10% based on the conformal mapping calculation presented by Ramer [12].

The observation of a decreasing current when both the fundamental and SH light are propagating through the waveguide, and an increasing current when only fundamental or only SH light is sent through strongly supports the SHG models which rely on the buildup of a DC field in the glass. When no light is present, a current was observed to flow due to the charging of the electrodes by the electrometer. When a combination of green and IR is present, the buildup of an internal electric field in the glass causes a current to flow in the electrodes which opposes the existing current and the total current is observed to decrease.
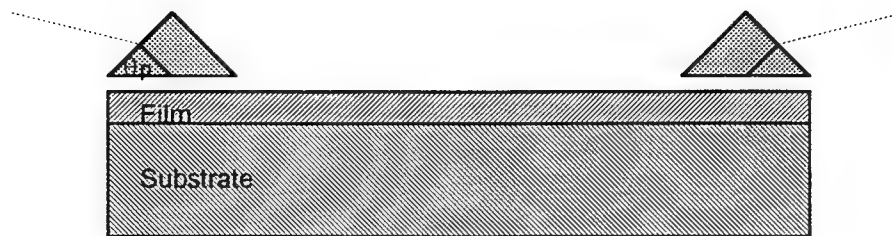
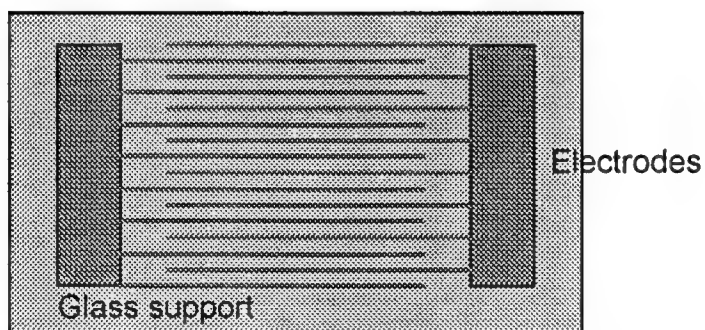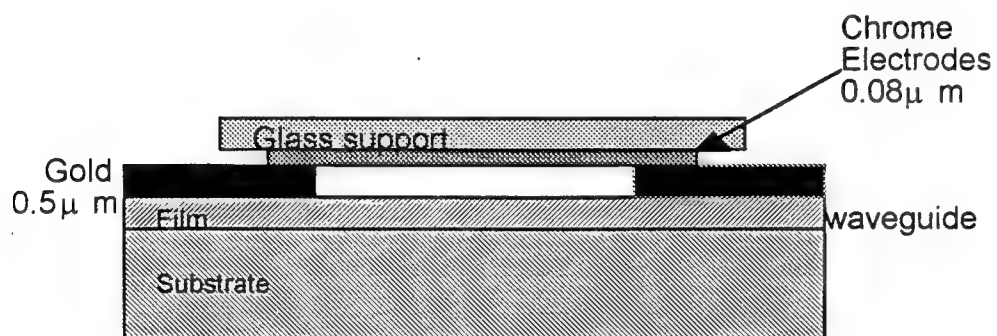**Figure 1. Planar waveguide geometry with prism coupling**

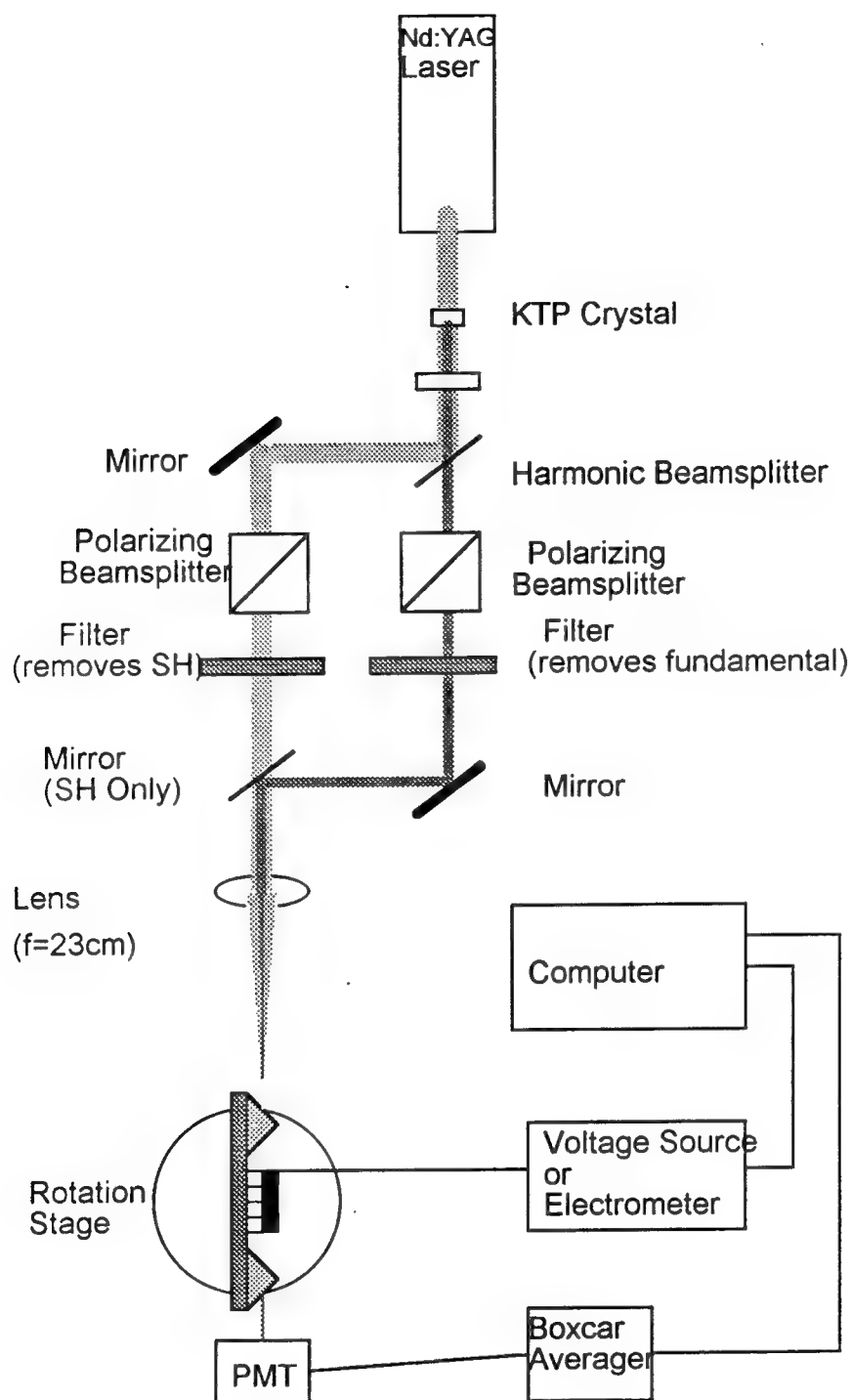**Figure 2. Planar waveguide with periodic electrode structure**

Figure 3. Experimental setup

Figure 4. Generated second harmonic as a function of applied voltage. The input IR is in the TM_0 mode and the output green is detected in the TM_0 mode.
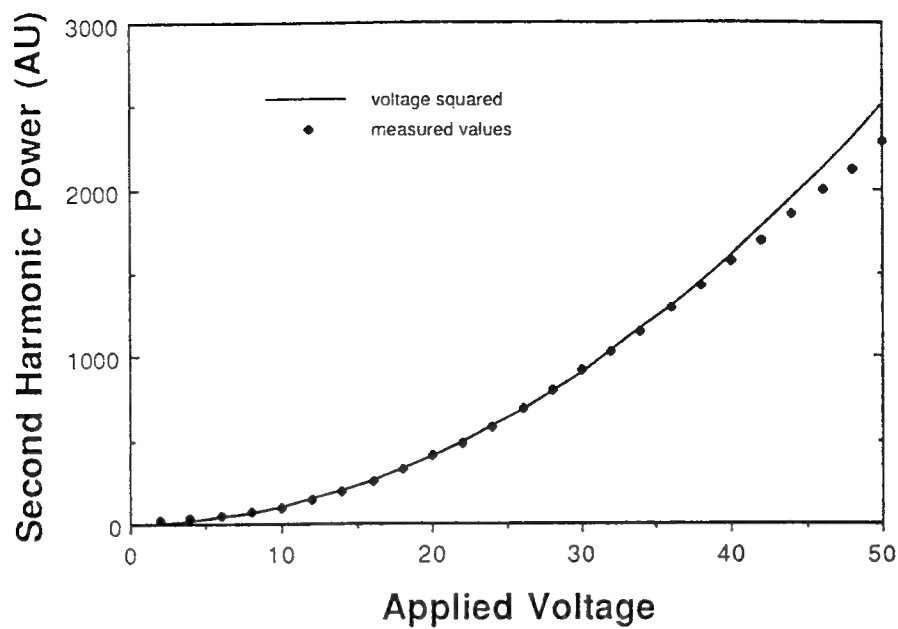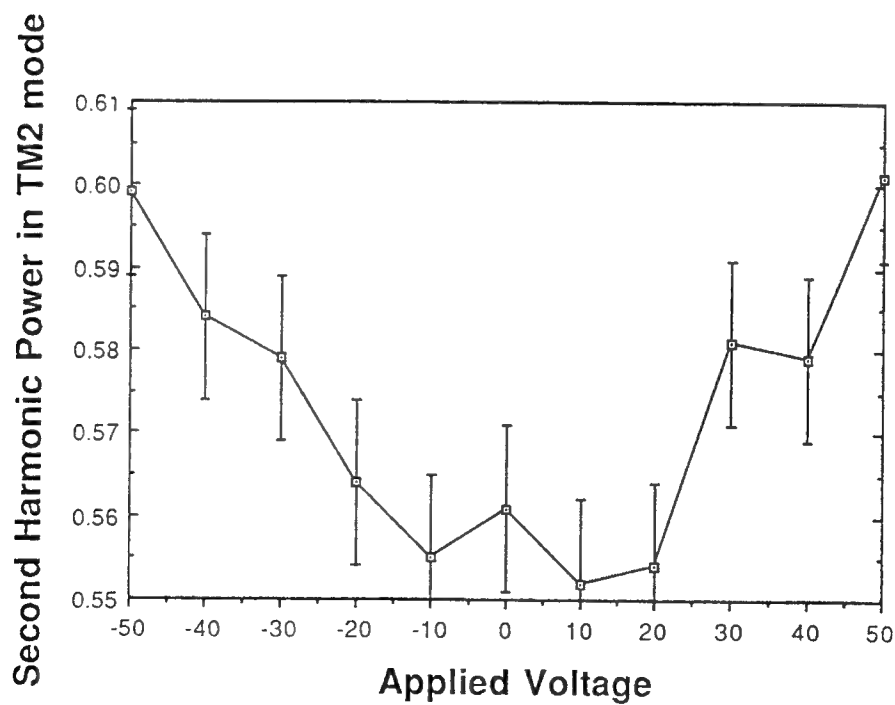

Figure 5. Generated second harmonic as a function of applied voltage. The input IR is in the TM_0 mode and the output green is detected in the TM_1 mode.
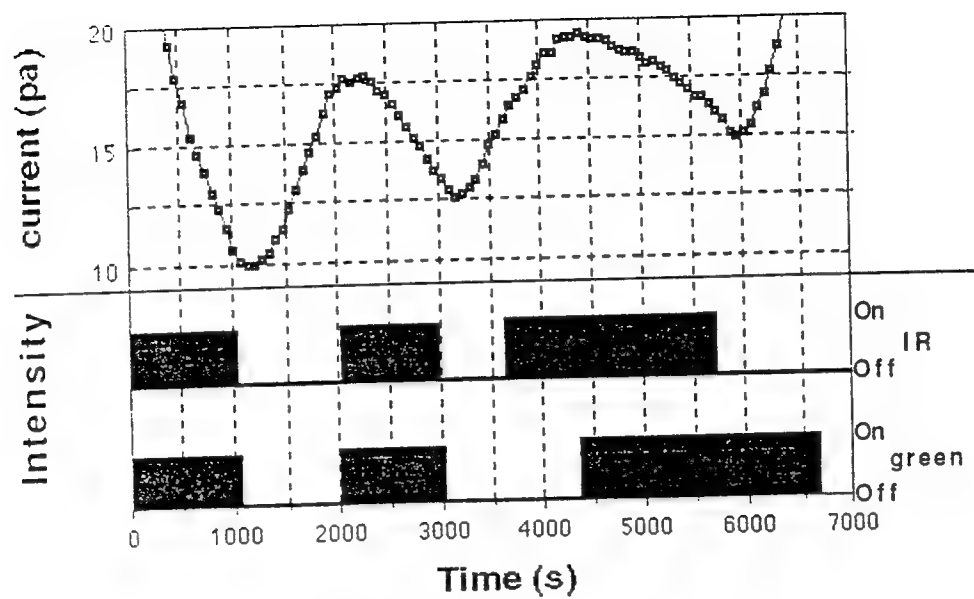
15-11

Figure 6. Effect of coupling in fundamental, SH, and both simultaneously on the current induced in the periodic electrode structure.
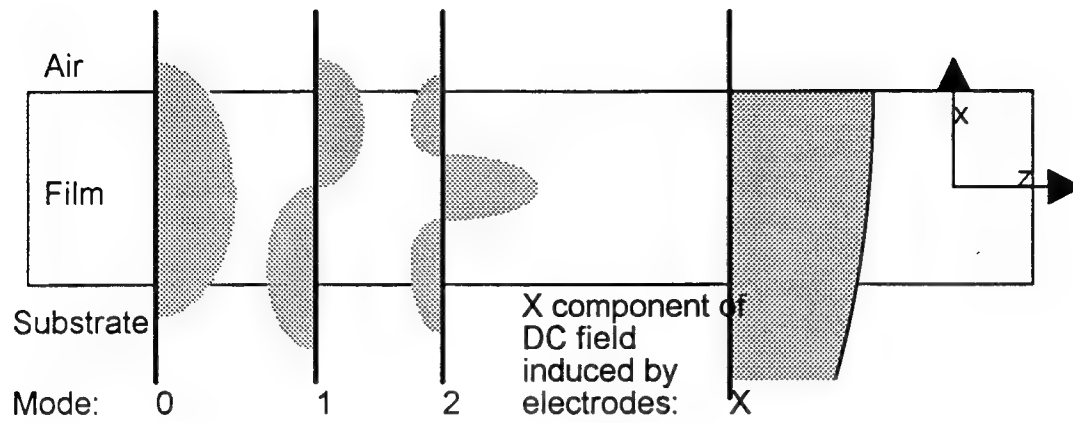
Figure 7. Electric field distributions in the waveguide for optical and applied DC fields.

1. U. Osterberg and W. Margulis, Opt. Lett. **11**, 516 (1986).

2. J. J. Kester, P. J. Wolf and W. R. White, Opt. Lett **17**, 1779 (1992).

3. D. Z. Anderson, V. Mizrahi and J. Sipe, Proc. Soc. Photo-opt. Instrum. Eng. **1516**, 154 (1991).

4. E. M. Dianov, P. G. Kazansky and D. Y. Stepanov, Soviet Lightwave Communications **1**, 381(1991).

5. N. M. Lawandy, Optics Communications, **74** 180 (1989).

6. M. C. Farries, P. St. J. Russel, M. Fermann and D. N. Payne, Electronics Letters, **23**, 322(1987).

7. R. H. Stolen and H. W. K. Tom, Optics Letters, **12**, 585(1987).

8 B. F. Levine and C. G. Bethea, J. Chem. Physics, **63**, 2666(1975).

9. R. Kashap, J. Opt. Soc. Am. B, **6**, 313(1989).

10. B. E. A. Saleh and M. C. Teich, *Fundamentals of Photonics*, (John Wiley, New York, 1991).

11. D. M. Krol, M. M. Broer, K. T. Nelson, R. H. Stolen, H. W. K. Tom and W. Pliebel, Opt. Lett. **16**, 211(1991).

12. O. G. Ramer, IEEE J. Quant. Elec. **18**, 386(1982).

PARTIAL DEMINERALIZATION
OF CANINE MANDIBULAR CORTICES

Sanjay Jain

Medical Student

Northeastern Ohio Universities College of Medicine

State route 44

Rootstown, Ohio 44272-0095

Final Report for:

Graduate Student Research Program

Clinical Investigations Directorate

Wilford Hall Medical Center

Lackland AFB

Partial Demineralization of Canine Mandibular Cortices

Sanjay Jain
Medical Student
Northeastern Ohio Universities College of Medicine

Abstract

Partial demineralization is a technique used to process cortical bone alloimplants. Obtaining consistency in this procedure while maintaining a prescribed level of mechanical strength is the chief concern of this protocol. The canine mandible model was used in this study due to its cancellous property and application to mandibular reconstuction. Bone segments were demineralized with 0.6N HCl over a range of time to determine an appropriate time to attain a level of 23%-60% demineralization. A time of approximately $3\frac{1}{2}$ hours was established. A new set of samples were later demineralized in $3\frac{1}{2}$ hours to analyze for relative consistency. 92% of the segments achieved partial demineralization within a range of 23%-60%. Despite a linear relation with time and degree of demineralization, several other variables exist which may influence this process.

Partial Demineralization of Canine Mandibular Cortices

Sanjay Jain

Introduction

Cortical bones usually undergo a particular type of processing to prepare it as an alloimplant. Demineralization of bone grafts is a technique involved with cortical preparation that has been shown to help promote new bone formation. It has been theorized that growth factors from the edge of a defect are released into the extracellular matrix causing nearby mesenchymal cells to differentiate into osteoblasts and chondroblasts.[1] This osteoinduction, as defined by Urist[2], is further assisted by the osteoconducting nature of a graft which provides a scaffold into which the cells migrate. Demineralized bone enhances the osteoconduction by reducing the obstacles for migratory activity.[3] Completely demineralized bone, although more easily incorporated, is not a viable replacement for defects in areas requiring a relatively high degree of mechanical strength. Thus it is important to obtain a level of demineralization which can provide such mechanical capabilities.

Guo *et al*[4] address this issue by investigating partially demineralized bone and its affects on mechanical strength. Their protocol used cortical bone plates from the dog femora. Using HCl as their solvent, they subjected their specimens to various periods of demineralization. They were able to conclude that a partial demineralization of 23%- 40% of normal bone mineral content provided "reasonable mechanical strength and normal potential for osteoinduction". The average time it took to attain this level was between 6-10 hours.

Their study, however, does not take into account several possible variables involved with demineralization. These variables include whole specimen demineralization, cortical thickness, and biological differences. In

16-3

our study, our main concern was to establish a protocol that would allow for a consistent partial demineralization of bone while maintaining the prescribed level mechanical strength for internal fixation and biological function. By doing so, we were able to identify many variables which can affect bone demineralization.

In proceeding with this study, we chose to use the canine mandible model because of its cancellous property and its similarity to the healing and remodeling processes of the tibia.[5] It is the ideal model for mandibular discontinuity defects, where it is one of the most frequent sites in the maxillofacial region requiring bone grafts for reconstruction.[6]

Materials and Method

Preparation

11 mandibles were harvested, but not aseptically, from mature female mixed breed dogs that had been euthanized for use in surgical teaching laboratories. The collected mandibles were cleaned of soft tissue and had the teeth extracted. The cancellous bone was removed with a high speed bur cooled by a saline drip. Each hemimandible was cut into two segments of approximately 2.5cm. These segments were identified as either the left or right side and anterior or posterior segment from a particular dog. These segments were then wrapped in saline soaked sponges and stored in a -70 °C freezer to prevent dehydration.[7]

Bone Analysis

Bone segments were removed from storage and thawed to room temperature. These specimens were later weighed. Mineral content of all segments were determined by dual energy x-ray absorbtiometry (DEXA)[†]. A standardized scan of an area of .4 cm^2 was performed in the center of each segment. A scan of the

† The use of a Lunar DPX model was provided by The Southwest Foundation for Biomedical Research

entire bone segment was also taken for total bone mineral content. Using vernier calipers, the cortical thickness was measured at the anterior, posterior, and superior borders of the medial and lateral wall. A mean value was then determined for the 6 sites per segment.

Demineralization

Process (for parts I & II): Demineralization was achieved by determining the amount of acid needed to provide for the total weight of bone. A proportion of 100 ml of acid solution / gm of bone was used. The acid concentration was 0.6N HCl.[4] Each bone specimen used was held in a permeable plastic cartridge and placed in the acid solution. A constant circulation was maintained with a magnetic stirrer. After removal of the specimen at the appropriate time, demineralization was halted by rinsing with deionized water. Mineral content of the treated specimens were again determined by DEXA to determine the change in bone density.

Part I: The purpose of this procedure was to determine an appropriate time in which to demineralize bone to attain the prescribed range of 23%-40%. For this step, 5 dogs were used providing a total of 20 samples. Demineralization was performed at one-half hour intervals to a maximum of 5 hours.

Part II: Using the data obtained in Part I, we proceeded to demineralize bone at a constant time. 6 dogs were used for this procedure giving a total of 24 samples. One segment from each dog was randomly selected for partial demineralization. The corresponding segment from the opposite hemimandible was used as the control only. The remaining 2 segments from that particular dog (one from each hemimandible) underwent the opposite treatment from its counterpart on that side. Thus 12 specimens were demineralized while the 12 corresponding segments from the opposite hemimandible remained intact so that

16-5

each demineralized segment had a control segment from the same anatomic location on the opposite side.

Statistical Analysis

Data collected was compared using a paired "t" test. A linear regression line was determined for the whole bone scan in *part I*. Correlation coefficients were also determined for the data.

Results

A scan of the entire bone as well as a smaller portion of the bone (.4cm^2) showed a large variance in terms of demineralization vs. time in *part I* (see Table 1). The entire bone scan after demineralization within certain intervals showed a linear relationship as seen in Fig. 1. The Pearson correlation coefficient was -0.8460257 and r^2= 0.7157594. The bone scan involving the smaller, select area of bone had a wide array of demineralization with time. The Pearson correlation coefficient was 0.11325 and r^2= 0.1282. The data indicated that 3½ hours was a suitable time to demineralize bone to arrive at a BMD level close to 40%.

Using 12 randomly selected bone specimens for *part II*, all the specimens were subjected to a time of 3½ hours to determine relative consistency. 91.67% (11 of 12) of these bones fell close to a range of approximately 23%-60% while 41.67% (5 of 12) fell close to a range of approximately 23%-40% as determined by an entire bone scan by DEXA. Ths smaller scan of .4cm^2 showed a greater degree of variation similar to that described in *part I*. The comparison of cortical thickness[†] to demineralization showed no definitive relationship. There also was no evident relation of the particular bone piece with the demineralization.

† *The mean cortical thickness of all bone samples was 2.24 ± .11 mm, SD = .40*

Discussion

Using the canine mandible model, this study shows that partial demineralization to an expected percentage is dependent on many factors involving a high degree of variability. In our experiments, all samples of bone were relatively the same size at 2.5 cm despite studies of self repair at this length without a graft.[8] The hydrogen ions in the medium were always kept in excess of what was truly needed.

Urist et al[9] were able to quantitate demineralization at a specific time by varying acid concentrations. They calculated a ratio of 1g of compact bone to approximately 16.3 ml of 0.6N HCl by basing it on two assumptions: "(a) that the chemical reaction $Ca_{10}(PO_4)_6(OH)_2 + 14\ HCl = 7CaCl_2 + 3Ca(H_2PO_4)_2 + H_2O$ is applicable to bone mineral as a whole, and (b) that bone contains 70% mineral, all in the form of hydroxyapatite, $Ca_{10}(PO_4)(OH)_2$." They demonstrated that during the early stages of demineralization, the bone to acid ratio was very important. The more hydrogen ions which were available in the demineralization medium, the more the decalcification which would occur. In general, a ratio of 10 ml HCl/g of bone (6.0×10^{-3} H ions/g bone) partially demineralized bone to a level close to 50% at 6 hours. An extrapolation of their results indicate that to achieve a range of approximately 23% - 40% demineralization, 15 ml HCl/g bone at 6 hours would be sufficient.

Unlike the previous study which varied acid concentrations, the variable that we measured in accordance with demineralization was time. We used a shorter time interval than the conventional times used for demineralization to account for the thin cortical wall of the mandible. The data indicated a linear relationship between time and amount of demineralization. We were looking for a suitable time for demineralizatpion of mandibles so that 23% - 40% could be attained. From our study in *part I*, our preliminary conclusion was that 3½ hours was an appropriate time for partial demineralization at a higher concentration of 100 ml HCl/g bone.

A problem arises though when you standardize time because of other variables which play into account. One such variable is the surface area exposure. Guo et al[4] used bone plates for their demineralization. This would insure a consistent flow of HCl throughout the bone surfaces. However most implants are performed using larger, intact specimens. We decided to use the cortical cribs of the mandible as a whole specimen for true application. As a result, the data showed several inconsistencies that is due to the altered flow of the free ions to some of the areas of bone.

The thickness of the bone also plays an important part in the demineralization process. Upon harvesting of bone, a bur was used to eliminate the cancellous bone. However this creates a rigid surface level and thus variances in cortical thickness throughout the wall. This variance can be evidenced from a different BMD measurement from one area of bone scanned with another area of bone scanned by DEXA.

Biological variability plays a rather minor role in this study since we used similar dogs that were all young females. The calcium content of normal cortical bone in dogs is 231.2 mg/g bone.[4] However the rabbits used by Urist et al had a BMC of 270.7 mg/g bone.[9] Overall, though, the ratio of inorganic to organic content is 70% to 30%. Certain bones in the body may have a slightly higher mineral content depending on its functional adaptation.

Other factors which may have distorted the outcome was the resistance to diffusible ions due to the plastic cartridges, an incomplete ionization of acid in the medium, and an inaccurate bone scans due to remnants of teeth left on the mandible.

Throughout our trials, we had been using duel x-ray absorbtiometry (DEXA)[10] as a tool to measure bone mineral content. This instrument is usually used clinically for measuring skeletal density in relation to osteoporosis and

other disease processes. It is based on the measurement of the relative absorption of 2 different energy, x ray beams that pass through the specimen. As a result, high resolution images are provided with a discrete reading of the bone outline. After scanning an area of interest, DEXA can calculate both bone mineral content (BMC) (gm) and bone mineral density (BMD) (gm/cm^2). An example of bone scans by DEXA is shown in Fig. 2. Accuracy is excellent (1%) with a correlation between bone mineral content and dry weight of r=0.965 (p< .0001).[11]

Conclusion

Due to the nature of the thin cortical bone of the mandible versus the femur, it is best to maintain a slightly higher level of BMC in order to provide enough strength for internal fixation. A broader range of partial demineralization involving 23%-60% is recommmended in comparison to the range of 23%-40% determined for the femur although further studies should be done involving strength tests to actually test this. The disadvantage of decreased demineralization is that there would be a slower rate of incorporation of the alloimplant with a minimal new bone yield. There are also many other variables involved that affect the process of demineralization that must be accounted for. A time range can be used as a standard for demineralization as we have been able to establish 3½ hours as an appropriate time for mandibles. A standard method can also be achieved by varying acid concentrations. In general, due to the differences in various bones, a preliminary analysis of partial demineralization should be performed on similar samples before demineralizing actual samples.

Bibliography

1. Hollinger, Jeffrey O., Kleinschmidt, James C.: The Critical Size Defect as an Experimental Model To Test Bone Repair Materials. *J Craniofacial Surg* 1:1, Jan 1990.

2. Urist, M.R.: Bone Transplants and Implants. *In* Urist,
   M.R.(ed.):Fundamental and Clinical Physiology of Bone. Philadelphia,
   J.B. Lippincott Co., 1980, p.331.

3. Nade, S.,Burwell, R.G.: Decalcified Bone as a Substrate For Osteogenesis.
   An appraisal of the interrelation of bone and marrow in combined grafts.
   J Bone Joint Surg. 59B 1977, 189-196.

4. Guo, M.Z., Zhen, X.S., Lin, L.B.: The Mechanical and Biological Properties
   of Demineralised Cortical Bone Allografts in Animals. *J Bone Joint
   Surg[Br]* 73B 1991, 791-794.

5. Najjar, T.A., Kahn, D.: Comparative Study of Healing and Remodeling in
   Various Bones. *J Oral Surg.* 35:375, 1977.

6. Kelly, J.K.: Maxillofacial Missle Wounds: Evaluation of Long-Term Results
   of Rehabilitation and Reconstruction. *J Bone Joint Surg.* 31:438, 1973.

7. Friedlander G.E.: Current Concepts Review Bone Banking. *J Bone Joint Surg*
   64A:2 1982.

8. Schmitz, J.P., Hollinger, J.O.: The Critical Size Defect as an
   Experimental Model for Craniomandibular Nonunions. *Clin. Orthop.* 205
   April 1986, 31-40.

9. Urist M.R., Strates B.S.,: Bone Formation in Implants of Partially and
   Wholly Demineralized Bone Matrix. *Clin Orthop* 71:271-8, 1970.

10. Cullum I.D., Ell P.J., Ryder J.P.: X-Ray Dual Photon Absorbtiometery: A
    New Method for the Measurement of Bone Density. *Brit J Radiology*
    62(739):587-592, 1989.

11. Gluer C.C., et al: Comparative assessment of Dual-Photon Absorbtiometry
    and Dual-Energy Radiography. *Radiology* 174:223-228, 1990.

Table 1. % Change in Bone Mineral Density (BMD) at Variable Times			
Time (hrs)	(%) Full Bone Scan	(%) Partial Bone Scan (.4 cm sq)	Mean Cortical Thickness (mm)
0.5	1.000	3.060	1.800
0.5	0.900	1.100	2.100
1.0	0.840	1.020	1.930
1.0	0.855	1.020	1.930
1.5	0.738	0.551	1.715
1.5	0.752	0.519	1.830
2.0	0.665	0.740	1.970
2.0	0.710	0.776	2.095
2.5	0.909	1.010	2.000
2.5	0.324	0.159	2.350
3.0	0.838	0.919	1.440
3.0	0.594	0.495	1.710
3.5	0.436	0.541	1.950
3.5	0.502	0.478	2.120
4.0	0.313	0.127	2.000
4.0	0.424	0.421	2.020
4.5	0.187	0.000	1.735
4.5	0.532	0.478	1.930
5.0	0.306	0.245	1.830
5.0	0.279	0.171	1.870

Table 2. % Change in Bone Mineral Density (BMD) at 3.5 Hours		
Mean Cortical Thickness (mm)	(%) Partial Bone Scan	(%) Full Bone Scan
1.27	0.240	0.339
1.87	0.096	0.237
1.95	0.486	0.587
2.03	0.402	0.482
2.08	0.429	0.357
2.10	0.222	0.507
2.27	0.571	0.611
2.27	0.160	0.222
2.32	0.556	0.781
2.58	0.218	0.350
2.67	0.369	0.433
2.77	0.367	0.464

PARTIAL DEMINERALIZATION
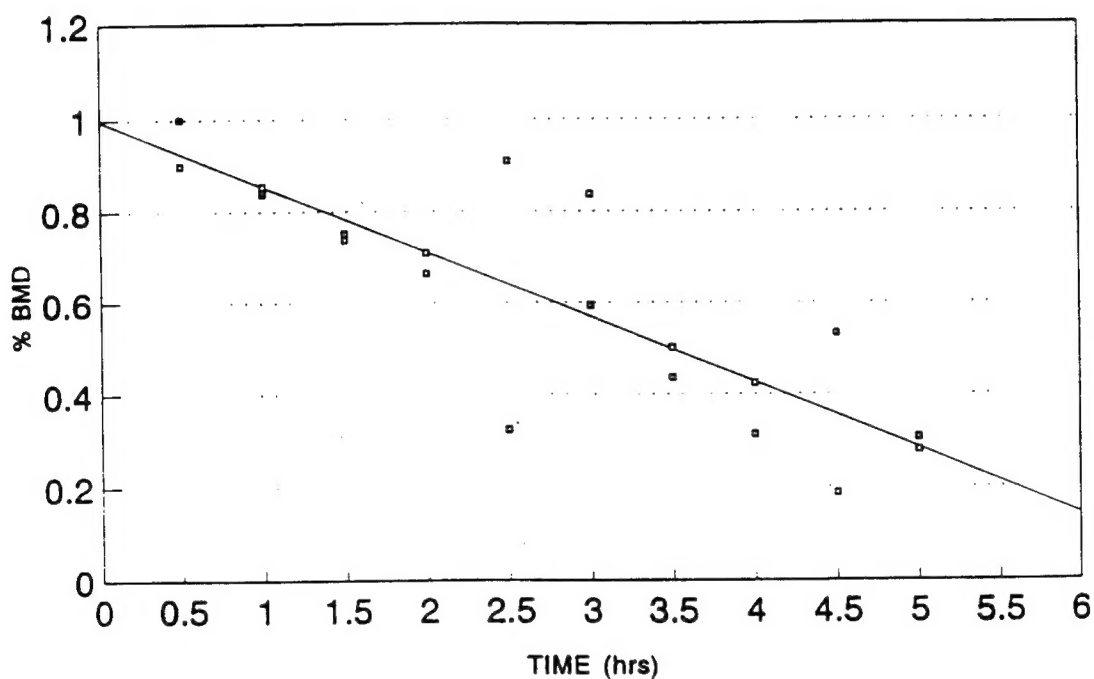## Time vs Percentage of BMD

Fig. 1. Demineralization of bone analyzed in its entirety by DEXA.
The linear regression line was defined by $y=ax+b$ where $a=0.9976$ and $b=-0.1426909$.
($n=20$, where $p < .0001$)

```
ID: 9005 LP, WHMC K-9   SCAN DATE: 07/28/93

        Manual  Analysis*

# CENTER   Number Of    BMC    AREA    BMD
    x,y     Vertices     g      cm²    g/cm²
1  43, 29      4         1.712   3.5    0.484




#   x ,  y
1  29.00, 12.00 a for research only

                ALTER
↑→ Move        + add vertex  Ctrl →←
←↓ Vertex      - del vertex  Select Vertex
           CtrlPgUp/CtrlPgDn-Size
```

LUNAR®

```
F1 - Move       F4 - Add ROI      F7 - Default ROIs   PgUp/PgDn-Select
F2 - Rotate     F5 - Delete ROI                       Esc - Next
                F6 - Analyze                          Home- Previous
```



```
ID: 9005 LP D/3.5, WHM  SCAN DATE: 07/30/93



# CENTER   Number Of    BMC    AREA    BMD
    x,y     Vertices     g      cm²    g/cm²
1  49, 38      4         0.859   3.5    0.243




#   x ,  y
0  63.00, 20.00 a for research only

                ALTER
↑→ Move        + add vertex  Ctrl →←
←↓ Vertex      - del vertex  Select Vertex
           CtrlPgUp/CtrlPgDn-Size
```

LUNAR®

```
F1 - Move       F4 - Add ROI      F7 - Default ROIs   PgUp/PgDn-Select
F2 - Rotate     F5 - Delete ROI                       Esc - Next
                F6 - Analyze                          Home- Previous
```

Fig. 2. The DEXA bone scans. The top reading is before demineralization and the bottom reading is after 3.5 hours of demineralization.

16-13